

Metropolia Ammattikorkeakoulu
Automaatiotekniikan koulutusohjelma

Ismo Leszczynski

Prediktiivinen pH:n säätö

Insinööritö 8.6.2010

Ohjaaja: lehtori Jukka Pirinen

Ohjaava opettaja: lehtori Jukka Pirinen

Tekijä Otsikko	Ismo Leszczynski Prediktiivinen pH:n säätö
Sivumäärä Aika	54 sivua 8.6.2010
Koulutusohjelma	automaatiotekniikka
Tutkinto	insinööri (AMK)
Ohjaaja Ohjaava opettaja	lehtori Jukka Pirinen lehtori Jukka Pirinen
<p>Insinööriityön tavoitteena oli tutustua prediktiiviseen pH-säätöön. Työn taustalla oli ajatus uuden säätötekniikan oppimisympäristön luomisesta Metropolia Ammattikorkeakoulun automaatiotekniikan laboratorioon. Tämä insinööriityö pohjustaa laboratoriotyön säätöosuutta. Työssä tutkitaan lineaarisen malliprediktiivisen säädön soveltuvuutta tehtävään ja tuloksia verrataan perinteiseen PID-säätimeen.</p> <p>Työssä luotiin Simulink-ympäristössä kaksi simulaatiota, joiden avulla analysoidaan Dynamic Matrix Control- ja PID-säätimien toimintaa yksinkertaisessa pH-prosessiohjauksessa. Säätöpiirien toimintaa tutkitaan eri malleilla, parametreilla, toimintapisteillä ja häiriöillä.</p> <p>Tehtyjen kokeiden perusteella voidaan todeta, että molemmat säätimet pystyvät suoriutumaan pH-säädöstä oikein viritettyinä. Säätimien välisessä vertailussa DMC-säädin suoriutui selvästi paremmin, mikä johtuu sen ennustavasta toiminnasta. Testeissä korostunut ero kasvaa edelleen, kun säädetään oikeaa pH-prosessia simulaation sijaan. Oikeissa prosesseissa säädinten haasteeksi tulee vielä prosessin kuollut aika, jonka huomioimiseen DMC-säätimellä on paremmat edellytykset kuin PID-säätimellä.</p>	
Hakusanat	pH, pH-mittaus, prediktiivinen pH:n säätö, MPC, DMC, PID

Helsinki Metropolia University of Applied Sciences Abstract

Author Title	Ismo Leszczynski Predictive pH control
Number of Pages Date	54 8 June 2010
Degree Programme	Automation Technology
Degree	Bachelor of Engineering
Instructor Supervisor	Jukka Pirinen, Senior Lecturer Jukka Pirinen, Senior Lecturer
<p>The target of this Bachelor's thesis was to examine predictive pH control methods. The motivation for this work was the need to study possible controller designs for a future laboratory process in the automation laboratory of Metropolia University of Applied Sciences. The chosen controller types studied in this thesis are a linear Model Predictive Controller based on the Dynamic Matrix Control design and a standard PID controller.</p> <p>The control designs in this Bachelor's thesis were tested using a simple pH process created in Matlab's Simulink. The process control was simulated using different models, parameters, measurement points and error signals.</p> <p>Based on the simulation tests it was concluded that both controller types were able to control the pH value relatively well. However, out of the two controller types the DMC controller clearly handles the changes faster. The difference in the performance of the controllers becomes clearer in actual pH processes with dead time and control restraints, which are handled better in the DMC design in comparison to the simple PI controller.</p>	
Keywords	pH, pH measurement, predictive pH control, MPC, DMC, PID

Sisällys

Tiivistelmä

Abstract

Symboli- ja lyhenneluettelo

1	Johdanto	8
2	Hapot ja emäket	9
	2.1 Liuoksen happamuus ja pH	11
	2.2 Puskuriliuokset	12
3	pH:n mittaus	13
	3.1 Lasielektrodi	13
	3.1.1 Lasikalvon resistanssi	15
	3.1.2 Asymmetriapotentiaali	16
	3.1.3 Alkali- ja happovirhe	16
	3.1.4 Vertailuelektrodi	17
	3.2 Mittauspiiri	17
	3.2.1 Mittalaitteiston kalibrointi	17
4	PID-säädin	18
	4.1 PID-säätimen viritys	19
	4.1.1 Ziegler-Nichols-menetelmä	19
	4.1.2 Cohen-Coon-menetelmä	20
	4.2 Prediktiivinen PID	20
5	Model Predictive Control	22
	5.1 Perusidea	22
	5.2 Kustannusfunktiot	24
	5.3 Mallit	25
	5.3.1 Askelvastemalli	25
	5.3.2 Impulssivastemalli	26
	5.4 Dynamic Matrix Control	27
	5.4.1 Viritysparemetrien vaikutus	31

6	Mallin vaikutuksen testaaminen	33
6.1	Toimintapiste	33
6.2	Askelvastemallin askel	34
6.2.1	Liian pieni askel	34
6.2.2	Liian suuri askel	35
6.2.3	Optimaalinen askel	36
7	pH-säätöpiirin toteutus	37
7.1	pH-prosessi	37
7.2	DMC-säädin	38
7.3	PI-säädin	38
7.4	DMC- ja PI-säätimien vertailu	39
7.4.1	Testi 1: Asetusarvomuutokset	39
7.4.2	Testi 2: Häiriökorjaus	42
7.5	Tulokset	43
8	Yhteenveto	44
	Lähteet	45
	Liitteet	
	Liite 1: pH-prosessin S-funktio ja ulostulon pH-muutosfunktio	46
	Liite 2: DMC: S-funktio ja ohjausmuutoksen laskentafunktio	47
	Liite 3: Step-response-simulaation Simulink-kaavio	49
	Liite 4: PI-säätimen koodi	50
	Liite 5: DMC- ja PI-säätöpiirien Simulink-kaaviot	51
	Liite 6: Asetusarvotiedosto	52
	Liite 7: SAA-mallinmuodostusfunktio	53
	Liite 8: Alkuarvotiedosto	54

Symboli- ja lyhenneluettelo

Kemia

A^-	happoa vastaava emäs
$a_{H_3O^+}$	oksoniumionien aktiivisuus
Ag	hopea
AgCl	hopeakloridi
B	emäs
$c_{H_3O^+}$	oksoniumionikonsentraatio
c_{OH^-}	hydroksidi-ionikonsentraatio
Cl^-	kloridi-ioni
E	jännite
F	Faradayn vakio
$\gamma_{H_3O^+}$	oksoniumionien molaarinen aktiivisuus
H^+	vetyioni
HA	happo
HB^+	emästä vastaava happo
HCl	vetykloridi
H_2O	vesimolekyyli
H_3O^+	oksoniumioni
M	moolimassa
Na^+	natriumioni
NH_3	ammoniakki
NH_4^+	ammoniumioni
OH^-	hydroksidi-ioni
R	yleinen kaasuvakio
T	absoluuttinen lämpötila

Säätötekniikka

d	häiriö
E	vapaa vaste
E^c	korjatut virhe-ennusteet
$e(t)$	prosessin asetusarvon ja ulostulon erotus hetkellä t
h	impulssivastekerroin
k	näytteenottohetki
K	vahvistusmatriisi
K_D	derivointivahvistus
K_I	integrointivahvistus

K_P	ohjausvahvistus
K_U	K_P oskillointipisteessä
L	prosessin kuollut aika
M	säätöhorisontti
N	ulostulon prosentuaalinen muutos (Cohen-Coon-menetelmässä)
N	mallin pituus
P	sisääntulon prosentuaalinen muutos (Cohen-Coon-menetelmässä)
P	ennustehorisontti
P_U	oskillointiaika
Φ	kustannusfunktio
R	prosessin kuollut aika jaettuna muutosajalla
r	asetusarvo
s	askelvastekerroin
S_f	dynaaminen matriisi, <i>future</i>
S_p	matriisi, <i>past</i>
t	näyteaika
T_D	derivointiaika
T_I	integrointiaika
Δu	ohjausmuutos
u	ohjaukset
$u(t)$	laskettu ohjaus hetkellä t
w/W	ohjausmuutoksen painokerroin
y	prosessin mitattu ulostulo
\hat{y}	mallin ulostulo
\hat{y}^c	korjattu mallin ulostuloennuste
\hat{Y}^c	korjatut ennustetut ulostulo

Lyhenteet

PID	Proportional-integral-derivative
MPC	Model Predictive Control
DMC	Dynamic Matrix Control
SISO	single input - single output

1 Johdanto

pH-säätö on monilla prosessitekniikan aloilla yksi tärkeimmistä säädöistä. pH-arvo vaikuttaa monissa prosesseissa niiden toimivuuteen, esimerkkeinä vedenpuhdistus, biokemialliset reaktiot ja mineraalien rikastus. Ongelmia säätöön tuottaa epälineaarisuus sekä viive ohjauksen ja mittauksen välissä.

Tämän työn tarkoituksena on tutustua pH-arvoon, sen mittaukseen ja säätöön. Tavoitteena on luoda pohja tulevaa säätötekniikan oppimisympäristöä varten Metropolia Ammattikorkeakoulun automaatiotekniikan laboratorioon.

Työssä tutkitaan PI- ja DMC-säätimien toimivuutta yksinkertaisen pH-prosessin ohjauksessa. Prosessit luodaan Matlabin Simulink-ympäristöön. Luoduille säätöpiireille tehdään askelmuutos- ja häiriökoe, joiden tuloksia verrataan aiheesta tehtyjen tutkimusten tuloksiin.

Työ on jaettu kahteen osaan. Ensimmäisessä osuudessa käsitellään pH:ta, sen mittausta ja eri säätömenetelmien teoriaa. Jälkimmäisessä osiossa esitellään säätötoteutus, sen toiminta ja tulokset.

2 Hapot ja emäkset

Hapon ja emäksen määritelmien mukaan hapot ovat protonin eli vetyionin (H^+) luovuttajia ja emäkset protonin vastaanottajia. Happoja, joilla on voimakas taipumus luovuttaa protoneja, kutsutaan *vahvoiksi* hapoiksi. *Heikoilla* hapoilla puolestaan on ainoastaan heikko taipumus protonien luovuttamiseen. Vastaavasti emäksiä, joilla on voimakas taipumus sitoa protoneja, kutsutaan vahvoiksi emäksiksi. Heikoilla emäksillä on heikko kyky sitoa protoneja. [5; 8; 12]

Jokaista happoa (HA) vastaa tietty protonin siirtyessä syntyvä emäs (A^-) ja päinvastoin. Näiden toisiaan vastaavien hapon ja emäksen, ns. happo-emäsparin HA/A^- , välillä pätee yhtälö



Protoni eli vetyioni on muita ioneja huomattavasti pienempi, ja sillä on tehokas vuorovaikutus dipolirakenteisen vesimolekyylien (H_2O) kanssa. Kun happoa (HA) liuotetaan veteen, sanotaan vetyionin hydratoituvan, jolloin liuokseen syntyy *oksoniumioneja* (H_3O^+). Luovuttaessaan protonin vesimolekyyllille happo synnyttää sitä vastaavan emäksen (A^-). Hapon ja veden välistä protoninsiirtoa kuvaa yhtälö:



Vastaavasti, kun emästä (B) liuotetaan veteen, syntyy liuokseen emästä vastaavaa happoa (HB^+) ja *hydroksidi-ioneja* (OH^-). Emäksen ja veden välistä protoninsiirtoa kuvaa yhtälö



Esimerkkinä vahvan hapon, vetykloridin (HCl), reaktio veden kanssa voidaan esittää kahdessa vaiheessa:



Vetyionit eivät kuitenkaan esiinny missään liuoksessa sellaisenaan, vaan aina sitoutuneena muihin ioneihin tai molekyyliin. Happo voi luovuttaa protonin ainoastaan, jos läsnä on emäs, joka voi ottaa sen vastaan. Edellä esitetyt osareaktiot on siis yhdistettävä yhdeksi reaktioksi [4]:



Esimerkkinä vahvan emäksen, ammoniakkin (NH_3) ja veden reaktio, jonka tuloksena syntyvät hydroksidi- (OH^-) ja ammoniumionit (NH_4^+):



Reaktiossa (5) kloridi-ioni on happoa (= vetykloridi) vastaava emäs ja reaktiossa (6) ammoniumioni on ammoniakkia vastaava happo, ns. ammoniakkin liittohappo.

Edellä vesimolekyyli toimi vetykloridin kanssa reagoidessaan emäksenä ja ammoniakkin kanssa reagoidessaan haponä. Tällaisia aineita, jotka olosuhteista riippuen toimivat joko happoina tai emäksinä, nimitetään *amfolyyteiksi*.

Happo-emäsreaktio tapahtuu vähäisessä määrin myös pelkkien vesimolekyylien välillä:



Tätä kutsutaan veden *autoprotolyysiksi*¹. Sekä oksonium- että hydroksidi-ionien väkevyydet ovat kuitenkin niin pienet, että puhdasta vettä pidetään sähköeristeenä.

¹ Kahden samanlaisen molekylin välistä protoninsiirtoreaktiota sanotaan autoprotolyysiksi.

Reaktion tasapainoasema on siis vahvasti vasemmalla. Tasapainotilassa jokaisen reaktioon osaaottavan aineen konsentraatio on vakio, joten H_3O^+ - ja OH^- -ionien konsentraatioiden tulo, ns. *veden ionitulo*, on sekä hyvin pieni että vakio. Mittausten mukaan veden ionitulo 25 °C lämpötilassa on $1,0 \cdot 10^{-14} \text{ (mol/l)}^2$. Veden autoprotolyysissä täytyy syntyneiden oksonium- ja hydroksidi-ionien määrien olla yhtä suuret, joten

$$c_{\text{H}_3\text{O}^+} = c_{\text{OH}^-} = \sqrt{1,0 \cdot 10^{-14} \text{ (mol/l)}^2} = 1,0 \cdot 10^{-7} \text{ mol/l.} \quad (8)$$

2.1 Liuoksen happamuus ja pH

Liuoksen happamuuden mitta on sen oksoniumionikonsentraatio, $[\text{H}_3\text{O}^+]$. Liuos on hapan, jos sen oksoniumionikonsentraatio on suurempi kuin hydroksidi-ionien konsentraatio $[\text{OH}^-]$. Päinvastaisessa tilanteessa liuos on emäksinen (alkalinen). [8]

Oksoniumionikonsentraatio vaihtelee yleensä välillä $1..1 \cdot 10^{-14} \text{ mol/dm}^3$.

Epäkäytännöllisen pienien lukuarvojen vuoksi happamuus ilmoitetaan logaritmisella asteikolla, jolloin pH-käsite määritellään seuraavasti:

$$\text{pH} = -\log[\text{H}_3\text{O}^+] \quad (9)$$

Näin määriteltynä liuoksen happamuutta kuvaava pH-arvo vaihtelee yleensä välillä 0-14. Kun liuoksen $\text{pH} < 7$, se on hapan. Kun liuoksen $\text{pH} > 7$, se on emäksinen. Neutraalin liuoksen, kuten esimerkiksi puhtaan veden, $\text{pH} = 7$. [5; 8; 12]

Vastaavasti voidaan hydroksidi-ionikonsentraatiosta laskea emäksisyys pOH:

$$\text{pOH} = -\log[\text{OH}^-] \quad (10)$$

Tämä on kuitenkin harvemmin käytetty laskutapa ja yleensä käytetään pH:ta.

Tarkemmin otettuna pH:n määritelmässä pitää oksoniumionikonsentraatio, $[H_3O^+]$, korvata sen aktiivisuudella $a_{H_3O^+}$:

$$pH = -\log a_{H_3O^+}. \quad (11)$$

$$a_{H_3O^+} \text{ on } [H_3O^+] \gamma_{H_3O^+}.$$

Kerroin $\gamma_{H_3O^+}$ voidaan arvioida määrittämällä liuoksen keskiaktiivisuus. [8]

Suhteellisen laimeille liuksille ($[H_3O^+] \leq 0,5 \text{ M}$) voidaan laskea pH-arvo käyttämällä laskuissa oksoniumionikonsentraatiota aktiivisuuden sijaan. Esimerkiksi vedelle ($pH = 7$) saadaan sama tulos kummallakin laskumenetelmällä. [8]

Liuoksen happamuudella tai emäksisyydellä on keskeinen merkitys kemiallisissa reaktioissa. Tietyt reaktiot vaativat tapahtuakseen korkean pH-arvon, kun taas toiset reaktiot eivät voi lainkaan tapahtua voimakkaasti emäksisissä liuksissa. Monet kemialliset teollisuusprosessit tapahtuvat hyvin rajoitetulla pH-alueella. [5]

2.2 Puskuriliuokset

Happo- tai emäslisäys aiheuttaa liuksessa yleensä pH-arvon muutoksen, joka voi olla jopa useita pH-yksiköitä. Monissa käytännön tilanteissa, kuten tyypillisesti biokemiallisissa prosesseissa, liuoksen pH-arvon pitäisi kuitenkin olla lähellä tiettyä optimiarvoa, koska pienetkin pH-muutokset voivat aiheuttaa suuria vahinkoja. Liuosta, jonka pH-arvo muuttuu vain merkityksettömän pienessä määrässä, kun siihen lisätään kohtuullisesti heikkoa happoa tai emästä, kutsutaan *puskuriliuokseksi*. Monissa biologisissa systeemeissä on puskuriliuoksia (esim. veri), mutta niitä voidaan myös valmistaa heikoista hapoista ja niitä vastaavista emäksistä. Puskuriliuksilla on tärkeä merkitys käytännön kemiassa; niitä lisätään sellaisiin liuksiin, joiden pH:n halutaan pysyä vakiona. [5; 8; 12]

3 pH:n mittaus

Liuosten pH:n mittaus on yksi tärkeimmistä prosessiteollisuuden mittauksista. Tarkoin säädetty pH on monien prosessien onnistumisen edellytys, ja toisaalta pH-tietoa tarvitaan myös laitteiden korroosiosuojauksen vuoksi [11].

Usein pH-säätöä suunniteltaessa mittauksen oletetaan olevan ideaalinen, jolloin mittauksen laatu jää helposti huomioimatta säätöä arvioitaessa. Käytännössä kuitenkin juuri mittauksen laadulla on merkittävin vaikutus säädön toimivuuteen, joten suunnitteluvaiheessa on erityisesti kiinnitettävä huomiota mittauksen laatuun. [12] pH-mittaus poikkeaa muista yleisistä prosessimittauksista siten, että sen ei voida olettaa toimivan optimaalisesti asennuksen jälkeen elinkaarensa loppuun asti. Se vaatii jatkuvaa kunnossapitoa, kalibrointia ja vikojen diagnosoimista. Myös pH-anturilla on prosessiriippuvainen elinkaari, vaikka huoltotoimenpiteet olisivat täydelliset koko käyttöajan. Seuraavassa esiteltävä lasielektrodi on pH-mittauksessa käytetyistä indikaattorielektrodeista yleisin. [8; 11; 12]

3.1 Lasielektrodi

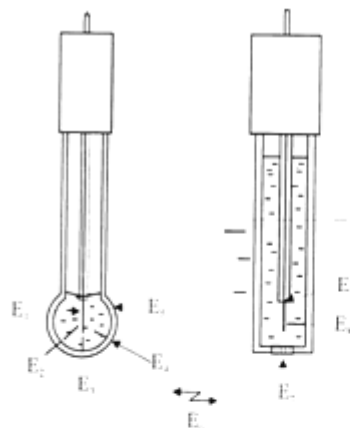
Lasielektrodi koostuu membraanista eli ohuesta erikoislasista valmistetusta lasikalvosta, sisäelektrodista ja täyttönesteestä. Täyttöneste on yleensä neutraalia eli sen $\text{pH} = 7$. Elektrodin sisäliuos on Cl^- -ioneja sisältävä puskuriliuos, jonka pH-arvon on pysyttävä muuttumattomana elektrodin toiminnan takaamiseksi. Sisäliuoksen puskurointi on välttämätöntä, koska lasista liukenee pieniä määriä alkalimetalli-ioneja liuokseen. Sisäelektrodina toimii yleensä hopea-hopeakloridielektrodi. [8; 11; 12]

Muita vaihtoehtoja sisäelektrodille ovat kalomeli- tai Thalamid-elektrodit. Thalamid-elektrodi soveltuu erinomaisesti korkeisiin lämpötiloihin lasielektrodin sisäiseksi vertailuelektrodiksi. [8; 12]

Lasielektrodiä ympäröivän lasikalvon koostumus vaikuttaa elektrodin selektiivisyyteen, pH-elektrodin happo- ja alkalivirheeseen, lasin kemialliseen kestävyys ja lasin resistiivisyyteen.

Lasikalvon paksuus on yleensä 50-200 μm , mutta vahvoja liuoksia mitatessa paksuus saattaa olla jopa 1 mm. Lasin rakenne on epäsäännöllinen kolmiulotteinen verkkorakenne, jossa piiatomit sitovat neljää happiatomia tetraedrisesti. Suurin osa happiatomeista on sitoutunut kahteen piiatomiin, mutta osa happiatomeista on sitoutunut vain yhteen piiatomiin muodostaen negatiivisia varausalueita. Lasissa olevat natrium- ja/tai litiumionit kompensoivat varauksellaan happiatomien negatiivisen varauksen ja toimivat samalla sähkövarausten kuljettajina.

Upotettaessa liuokseen lasikalvon pintakerroksen yli syntyy potentiaaliero (kuva 1), jonka arvo riippuu liuoksen pH:sta. Lasielektrodi pystyy indikoimaan liuoksen pH:n, koska elektrodin sisäliuoksen pH pysyy muuttumattomana. [6]



Kuva 1. Lasi- ja vertailuelektrodien osajännitteet sekä niiden muodostaman kennon kennojännite [6].

- E_1 = lasielektrodin AgCl:llä päällystetyn Ag-langan ja sisäliuoksen välisen liitoskohdan välinen jännite
- E_2 = Ag-langan ja lasielektrodin sisäpinnan välinen jännite
- E_3 = lasikalvon sisäpinnalla oleva jännite
- E_4 = lasikalvon yli vaikuttava jännite eli ns. asymmetriapotentiaali
- E_5 = lasielektrodin ulkopinnan ja mitattavan liuoksen välinen jännite
- E_6 = lasi- ja vertailuelektrodien välinen jännite eli mitattavan kennon kennojännite
- E_7 = nestepotentiaali eli ryömintäpotentiaali kohdassa, jossa vertailuelektrodi koskettaa mitattavaa liuosta
- E_8 = vertailuelektrodin sisäpinnan ja vertailuelektrodissa olevan pH-mittarille vievän metallilangan välinen jännite. Jännitteen suuruus riippuu elektrodin Cl^- -aktiivisuudesta
- E_9 = metallien liitoskohdan välinen jännite

Lasikalvon pinnan ja liuoksen välinen potentiaali noudattaa Nernstin yhtälöä eli potentiaaliero on suoraan verrannollinen vetyioniaktiivisuuden logaritmiin [8]:

$$E_3 = E^0 + \frac{RT}{F} \ln (a_{H^+})_x. \quad (12)$$

E^0 on normaali- eli standardipotentiaali

E_3 on lasikalvon sisäpinnalla oleva jännite

R on yleinen kaasuvakio

T on absoluuttinen lämpötila

F on Faradayn vakio

$(a_{H^+})_x$ on ulkoisen liuoksen vetyioniaktiivisuus.

Lasikalvon sisäpuolella oleva potentiaali E_2 muodostuu saman periaatteen mukaisesti kuin E_3 , tällä kertaa aktiivisuutena käytetään elektrodin sisäliuoksen aktiivisuutta, joka on vakio [8].

3.1.1 Lasikalvon resistanssi

pH-arvon määrittämiseen tarvitaan potentiaaliero mitta- ja vertailuelektrodeihin.

Lasikalvon läpi kulkee mittauksen aikana heikko sähkövirta, jota vastustaa lasin resistanssi. Lasikalvon resistanssi on yleensä noin 10-500 MΩ, ja se vaihtelee lämpötilan funktiona. Esimerkiksi Corning 015 -lasin resistanssi on huoneenlämmössä noin 200 MΩ, kun taas 10 °C lämpötilassa se on noin 1000 MΩ. [12]

Lasikalvon korkea resistanssi aiheuttaa käytännön ongelmia mittapiirien rakentamisessa. Ongelmien helpottamiseksi on kehitetty erityyppisiä lasia eri mittaustilanteisiin. Mitattavien liuosten happamuus ja lämpötila vaikuttavat suuresti valintaan, koska lasikalvon resistanssin pitää olla aina mahdollisimman pieni. [12]

3.1.2 Asymmetriapotentiaali

Asymmetriapotentiaali on vallitseva potentiaaliero lasikalvon sisä- ja ulkopintojen välillä. Asymmetriapotentiaali aiheuttaa virheen mittauksessa, jos sitä ei eliminoida kalibrointivaiheessa. Sen mittaus voidaan suorittaa siten, että lasikalvon sisä- ja ulkopuolella olevien liuosten koostumus on täysin sama. Asymmetriapotentiaalın suuruus on vain muutamia millivoltteja, mutta tyypillisesti potentiaali muuttuu elektrodin vanhetessa. Asymmetriapotentiaali voi aiheutua seuraavista syistä:

- Lasikalvon koostumus vaihtelee sisä- ja ulkopuolella.
- Kalvon ulkopinta on altis kemiallisille ja mekaanisille rasituksille.
- Valmistusvaiheessa elektrodin kuumentaminen vähentää alkalimetalli-ionien määrää kalvon ulkopinnalla.
- Kalvon pintojen hydratoituneet kerrokset eroavat toisistaan. Lasista saattaa liueta yhdisteitä, jotka jäävät kalvon sisäpuolelle. Ulkopuolen kerros ei sisällä näitä yhdisteitä, koska elektrodi puhdistetaan säännöllisesti.

Muuttuvan asymmetriapotentiaalın takia pH-mittari on kalibroitava säännöllisesti. [8; 12]

3.1.3 Alkali- ja happovirhe

Erittäin emäksisiä ja happamia liuoksia mitattaessa lasielektrodin käyttäytyminen poikkeaa ideaalisesta, eli suhde jännite/pH ei ole lineaarinen. Lasielektrodi toimii ideaalisesti normaalilämpötilassa pH-alueella 3-10. Jos lämpötila muuttuu normaalia korkeammaksi, alkaa toiminta poiketa ideaalisesta jo aiemmin. [8]

Voimakkaasti emäksisiä liuoksia mitattaessa saadaan todellista alhaisempi pH-arvo. Tämä johtuu emäsluoksessa olevista lasielektrodin toimintaa häiritsevistä alkalimetalli-ioneista, lähinnä Na^+ -ioneista. Esimerkiksi liuoksessa, joka on 1-molaarinen Na^+ -ionien suhteen, alkalivirhe pH-arvolla 12 on noin 1,1 pH-yksikköä. [8]

Voimakkaasti happamien liuosten tapauksessa mitattu pH on sen todellista arvoa suurempi. Lämpötilalla ei ole suurta merkitystä happovirheeseen. Happovirheen oletetaan johtuvan siitä, että lasikalvo absorboi happomolekyylejä, jolloin vetyioniaktiivisuus lasissa kasvaa. [8]

3.1.4 Vertailuelektrodi

pH-mittaus vaatii aina mittauselektrodin ohella vertailuelektrodin, joka antaa referenssipotentiaalin. Tuloksen saamiseksi mittauselektrodin potentiaalia verrataan referenssipotentiaaliin. Referenssipotentiaalin täytyy pysyä vakiona prosessiliuoksen ominaisuuksien muuttuessa. [12]

3.2 Mittauspiiri

Tyypilliseen pH-mittauspiiriin kuuluu mittauselektrodi, vertailuelektrodi ja mittausvahvistin. Elektrodit muodostavat piirin jännitelähteen ja mittausvahvistin vahvistaa pienitaajuisen signaalin käytettävälle tasolle. Useimmiten mittausvahvistin muuttaa signaalin suoraan pH-arvoksi. Vaihtoehtoisesti voidaan mitata jännitearvot ja laskea niitä vastaavat pH:t. [12]

3.2.1 Mittalaitteiston kalibrointi

Mittalaitteisto kalibroidaan käyttämällä muutamaa eri standardipuskuriliuosta. Näihin puskuriliuoksiin kuuluu aina liuos, jonka $\text{pH} = 7$, koska mittapiirin ulostulojännite on silloin teoriassa nolla volttia. Happoliuoksia mitattaessa kalibroinnissa käytetään myös puskuriliuosta, jonka happamuus on lähellä arvoa $\text{pH} = 4$, kun taas emäsluosten tapauksessa käytetään liuosta, jonka emäksisyys on lähellä arvoa $\text{pH} = 10$. Vaihtoehtoisesti kalibroinnissa voidaan käyttää kaikkia kolmea erilaista puskuriliuosta, mutta tämä ei ole kuitenkaan välttämätöntä. [12]

4 PID-säädin

Teollisuuden ehdottomasti käytetyin prosessisäädin on PID (Proportional-integral-derivative) -säädin. PID-säätimen suosio johtuu sen käytettävyyden yksinkertaisuudesta, joka mahdollistaa helpon säädettävyyden prosessin ohjaajille. Säätimen toiminta tunnetaan erittäin hyvin, mikä lisää sen suosiota suunnittelijoiden keskuudessa. PID-säätimillä on myös hyvä kohinan sietokyky. PID-säätimiä tuotetaan tuhansittain erilaisilla kokoonpanoilla, ja niillä saadaan hyvin toimiva ohjaus suurimmalle osalle prosesseista hyvin viritetyillä parametreilla. [7; 12; 13]

Perus-PID-säätimen laskentaa voidaan esittää seuraavalla kaavalla [12]:

$$u(t) = K_P \left(e(t) + \frac{1}{T_I} \int_0^t e(t) dt + T_D \frac{de(t)}{dt} \right). \quad (13)$$

K_P on ohjausvahvistus

T_I on integrointiaika

T_D on derivointiaika

$u(t)$ on laskettu ohjaus hetkellä t

$e(t)$ on prosessin asetusarvon ja ulostulon erotus hetkellä t

PID-säädin voidaan jakaa myös pienempiin kokonaisuuksiin. Esimerkkikokoonpanoja ovat P-, PI- ja PD-säätimet. Eri kokoonpanot on helppo luoda, algoritmista tulee vain poistaa haluttu osuus, niin saadaan haluttu kokoonpano.

P-säätö vastaa säädön muutosnopeudesta, joka riippuu erosuureen koosta. Pienellä erosuureella P antaa pienen ohjauksen ja suurella erosuureella suuren. P-säätö itsessään on altis virheille, koska se käyttää vain laskentahetkellä olevaa virhettä eikä arvioi virhehistoriaa. Tästä johtuu useimmiten P-säätimelle ominainen vakiovirhe. [12; 13]

P-säätimen virhettä voidaan korjata lisäämällä säätimeen I-osuus. Säätimen I-säätö vastaa virhehistorian integroinnista, jolloin I-säätö laskee tai kasvattaa ohjauksen suuruutta virheen poistamiseksi. I-säätö voi aiheuttaa oskillointia ja ohjauksen

hidastumista, mutta se poistaa tehokkaasti P-säätimen vakiovirheen ja yleisen prosessikohinan. [12; 13]

D-säätö tarkastelee erosuureen muutosnopeutta. Se pyrkii kompensoimaan poikkeamaa jo siinä vaiheessa, kun se alkaa muodostua. D-säädön käyttö on kuitenkin rajoitettua, sillä se vahvistaa kohinaa. Tästä syystä on oltava tarkkana säätökomponentteja valittaessa. [12; 13]

4.1 PID-säätimen viritys

PID-säätimen virittämiseen on monia eri menetelmiä. Säätimelle ei voida nimetä yleispäteviä parametreja, vaan ne täytyy virittää jokaiselle prosessille. PID-säädintä viritettäessä täytyy ottaa huomioon, että jonkin ominaisuuden parantaminen voi usein heikentää toista ominaisuutta. Jos esimerkiksi halutaan suurempi muutosnopeus prosessille, voi vahvistuksen lisääminen lisätä oskillointia. Toisaalta pienemmän vahvistuksen käyttäminen vähentää oskillointia, mutta myös hidastaa prosessin muutosta. [7; 13; 14]

PID-säätimen vitysmenetelmistä esittelen seuraavassa kaksi yleisintä, Ziegler-Nichols- ja Cohen-Coon-menetelmät, joista jälkimmäistä on käytetty tämän työn säätimen virittämiseen. [13; 14]

4.1.1 Ziegler-Nichols-menetelmä

Ziegler-Nichols-menetelmässä asetetaan aluksi säätimen I- ja D-osuuksien arvot nolnaan. Tämän jälkeen P-osuuden vahvistusta kasvatetaan, kunnes prosessi alkaa oskilloida. Tämän ns. kriittisen pisteen saavuttamisen jälkeen voidaan laskea P-, I- ja D-osuuksien vahvistusarvot taulukon 1 mukaisesti [14].

Taulukko 1. Ziegler-Nichols-menetelmän laskut

Ohjaustyyppi	K_P	K_I	K_D
P	$0,5K_U$		
PI	$0,45K_U$	$1,2 K_P/P_U$	
PID	$0,60K_U$	$2 K_P/P_U$	$K_P P_U/8$

K_P = suhteellinen vahvistus

K_I = integrointivahvistus

K_D = derivointivahvistus

K_U = K_P oskillointipisteessä

P_U = oskillointiaika

4.1.2 Cohen-Coon-menetelmä

Cohen-Coon-menetelmän parametrilaskenta suoritetaan prosessin tasapainotilassa suoritetusta askelvastekokeesta. Saadusta askelvastekuvaajasta katsotaan tarvittavat arvot ja lasketaan parametrit taulukon 2 mukaisesti [14].

Taulukko 2. Cohen-Coon-menetelmän laskut

Ohjaustyyppi	K_C	T_I	T_D
P	$(P/NL)*(1+(R/3))$		
PI	$(P/NL)*(0,9+(R/12))$	$L*(30+3R)/(9+20R)$	
PID	$(P/NL)*(1,33+(R/4))$	$L*(30+3R)/(9+20R)$	$4L/(11+2R)$

K_C = vahvistus

T_I = integrointiaika

T_D = derivointiaika

P = sisääntulon prosentuaalinen muutos

N = ulostulon prosentuaalinen muutos jaettuna muutosajalla

L = prosessin kuollut aika

R = prosessin kuollut aika jaettuna muutosajalla

4.2 Prediktiivinen PID

Prediktiivinen PID-säädin on kehittyneempi versio perus-PID-säätimestä.

Prediktiiviseen PID-säätimeen on sisällytetty monia Model Predictive Controllin (MPC) tärkeistä ominaisuuksista. Model Predictive Controllia tarkastellaan luvussa 5.

Prediktiivisen PID-säätimen toiminta perustuu P määrään rinnakkaisia PID-säätimiä, jossa P on ennustehorisontin pituus. Jokaisessa PID-säätimessä on samat viritysparametrit. Rinnakkaiset säätimet toimivat eri ajanhetkille ennustetuilla virheillä. [7]

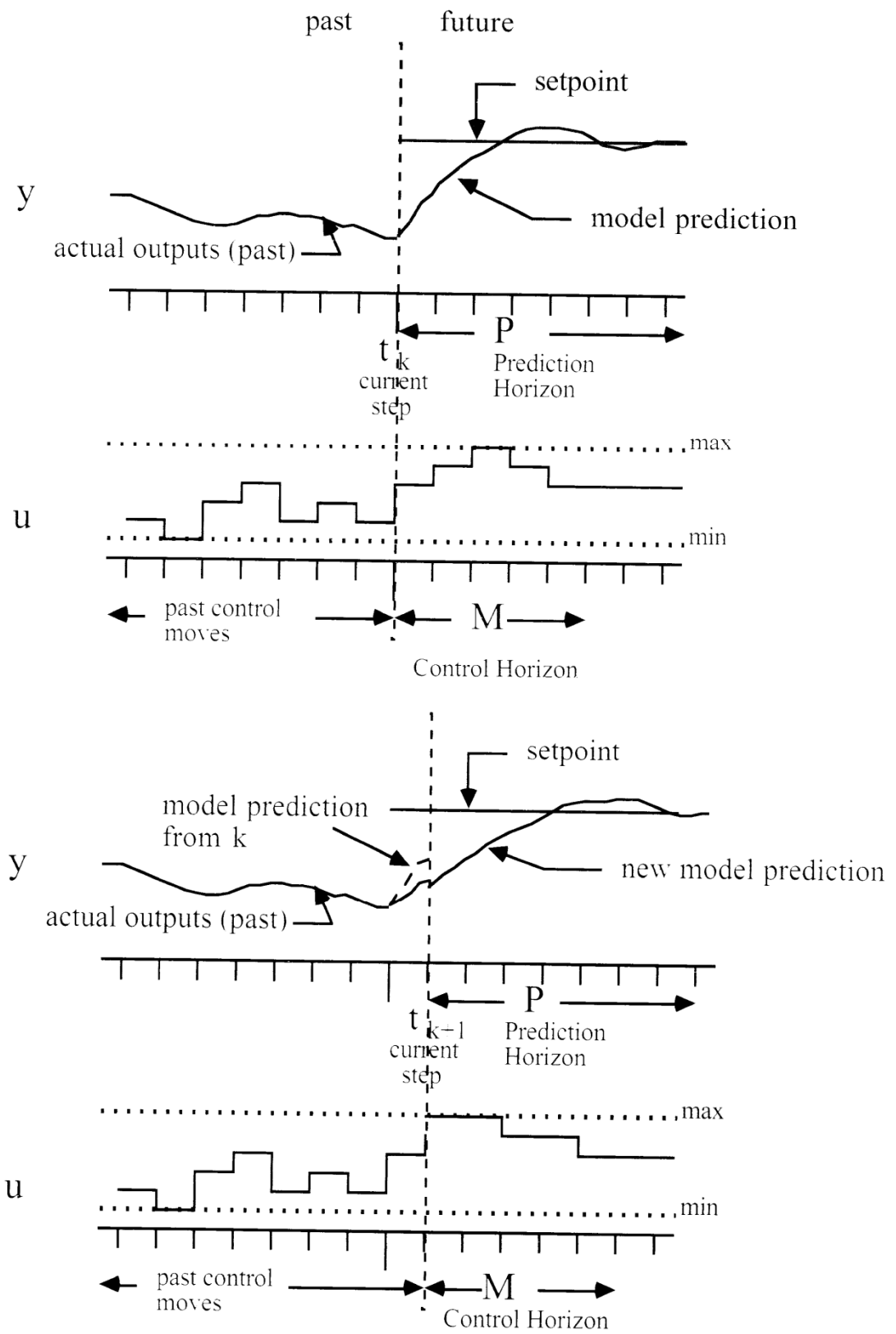
Prediktiivisen PID-säätimen vahvuus on käyttöönoton ja operaattorikoulutuksen helppous. Tyypillisissä prosessiympäristöissä suurin osa säädöistä on toteutettu PID-säätimillä, joita voidaan käyttää suoraan prediktiivisen säädön luomiseen. Tällöin ei tarvitse uusia koko prosessiohjausta ja operaattorien koulutusta. Vaihtoehtoisessa MPC-säädössä koko säätölaitteisto täytyy uusia ja operaattoreille pitää järjestää koulutusta uuden säätömenetelmän toimintaan saamiseksi. [7]

5 Model Predictive Control

Model Predictive Control (MPC) on kehittynyt säätömenetelmä, joka on laajalti käytössä prosessiteollisuudessa, kemianteollisuudessa se on jopa käytetyin menetelmä. MPC:n vahvin puoli on prosessirajoitteiden selkeä käsittely, ja monimuuttujaprosessien muodostaminen aikaviiveiden kanssa on suoraviivaista. Seuraavassa havainnollistetaan MPC-säätöä SISO (single input - single output) -sädöllä. [2; 3]

5.1 Perusidea

MPC:n tavoitteena on *kustannusfunktion* minimointi *ennustehorisontin* P yli ohjausmuutosten suhteen *säätöhorisontissa* M käyttämällä hyväksi prosessin dynaamista mallia. Toisin sanoen tarkoituksena on laskea optimaalinen ohjaussekvenssi, joka pyrkii minimoimaan kustannusfunktion säätöhorisontin aikana. Vaikka ohjausmuutoksia lasketaan monta, vain ensimmäinen ohjausmuutos otetaan käyttöön. Ohjausmuutoksen käytön jälkeen saadaan uusi mittaustulos prosessista, jonka jälkeen suoritetaan virheenkorjaus mittauksen ja mallin välillä. Virheen korjauksen jälkeen lasketaan taas P -horisontille ohjausmuutokset M -horisonttiin. [2; 3] Kuva 2 havainnollistaa MPC:n perusidea.



Kuva 1. MPC:n perusidea [2].

5.2 Kustannusfunktiot

Kustannusfunktioista yleisin on ”least-squares” eli pienimmän neliösumman menetelmä. Tässä menetelmässä kustannusfunktio on ennustettujen virheiden ja laskettujen ohjausmuutosten neliöiden summa. [2; 3]

Esimerkiksi kustannusfunktio Φ ennustehorisontilla 3 ja säätöhorisontilla 2 näyttää seuraavalta [2]:

$$\Phi = (r_{k+1} - \hat{y}_{k+1})^2 + (r_{k+2} - \hat{y}_{k+2})^2 + (r_{k+3} - \hat{y}_{k+3})^2 + w\Delta u_k^2 + w\Delta u_{k+1}^2 \quad (14)$$

r on asetusarvo

\hat{y} on prosessimallin ulostulo

k on näytteenottohetki

Δu on ohjausmuutos kahden peräkkäisen näytteenottohetken välillä

w on painokerroin ohjausmuutoksille.

Yleisesti kustannusfunktio Φ ennustehorisontille P ja säätöhorisontille M voidaan kirjoittaa muotoon [2]

$$\Phi = \sum_{i=1}^P (r_{k+i} - \hat{y}_{k+i})^2 + w \sum_{i=0}^{M-1} \Delta u_{k+i}^2 \quad (15)$$

Toinen mahdollinen kustannusfunktio voidaan muodostaa virheiden ja ohjausmuutosten itseisarvojen summana [2]:

$$\Phi = |r_{k+1} - \hat{y}_{k+1}| + |r_{k+2} - \hat{y}_{k+2}| + |r_{k+3} - \hat{y}_{k+3}| + w|\Delta u_k| + w|\Delta u_{k+1}|. \quad (16)$$

Tässä on käytetty samoja ennuste- ja säätöhorisonttiarvoja kuten edellisessä esimerkissä, ja tämän yleinen muoto on seuraava [2]:

$$\Phi = \sum_{i=1}^P |r_{k+i} - \hat{y}_{k+i}| + w \sum_{i=0}^{M-1} |\Delta u_{k+i}| \quad (17)$$

5.3 Mallit

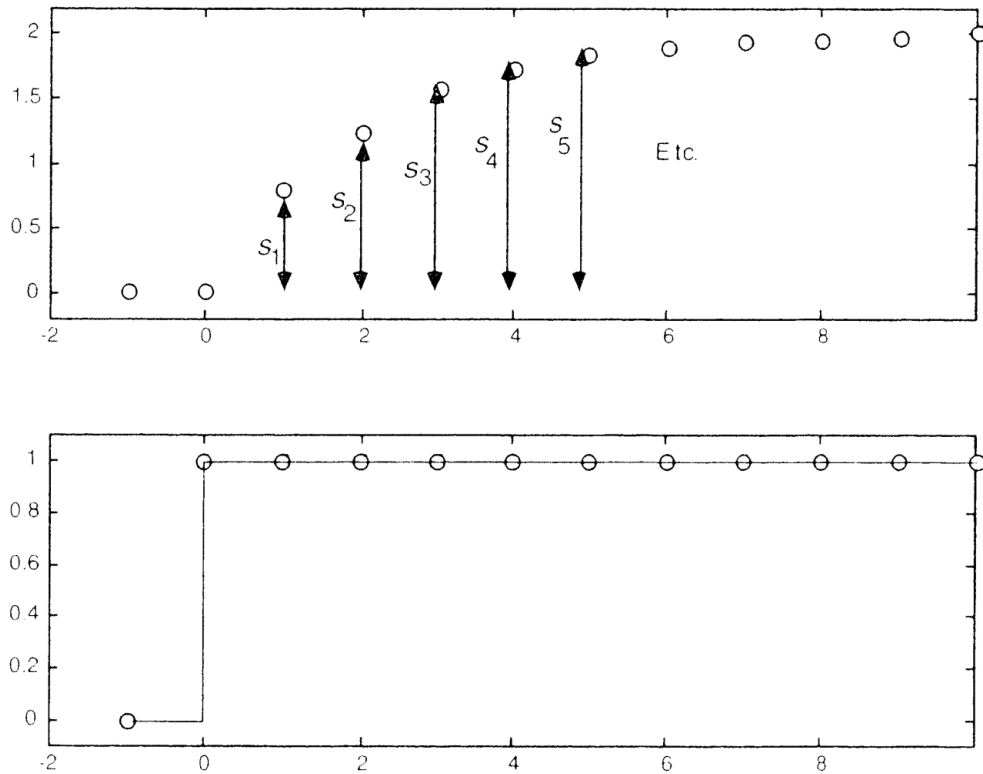
Prosessien mallintamiseen on olemassa monenlaisia mallityyppejä, joiden avulla voidaan arvioida prosessin ulostulo. Näiden mallien avulla voidaan ratkoa kustannusfunktioita. MPC on kehitetty digitaalisia eli aikadiskreettejä järjestelmiä varten, joten luontaisia mallityyppejä ovat askelvaste- ja impulssivastemallit, jotka kumpikin ovat hyvin yleisiä MPC-laskennassa. Askelvaste- ja impulssimalleissa on huomioitavaa, että ne soveltuvat vain open-loop-tyyppisiin prosesseihin ja vaativat huomattavan paljon parametreja verrattuna aika-avaruus- ja siirtofunktiomalleihin. [2; 3]

5.3.1 Askelvastemalli

Askelvastemallilla kuvataan prosessin reagoitua yksikköaskeleen suuruiseen muutokseen ohjauksessa. Malli saadaan kasvattamalla ohjausta yksikköaskeleella prosessin ollessa tasapainotilassa. Mallin kertoimet ovat yksinkertaisesti ulostulon arvot jokaisella aika-askeleella yksikköaskelmuutoksen jälkeen. Nämä kertoimet yhdistetään kerroinvektoriksi, jossa s_i on kyseisen i -näyteajan kerroin [2]:

$$S = [s_1 \ s_2 \ s_3 \ s_4 \ s_5 \ \dots \ s_N]^T \quad (18)$$

Kertoimien saantiperiaatetta on havainnollistettu kuvassa 3. Mallin pituus N valitaan siten, että sillä näytteenottohetkellä prosessi on suunnilleen tasaantunut.



Kuva 2. Askelvastemalli [2]

5.3.2 Impulssivastemalli

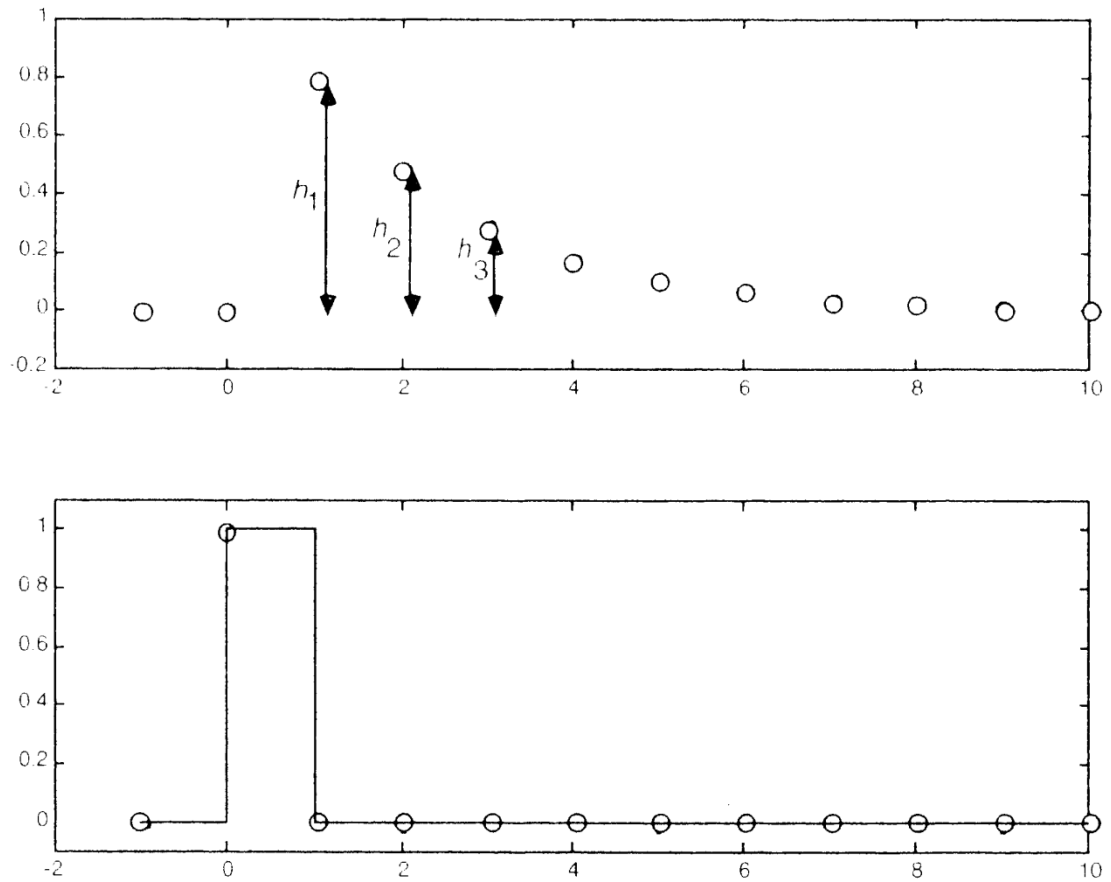
Askelvasteenmallin tavoin impulssivastemallin kertoimet saadaan muuttamalla ohjausta, mutta yksikköaskelmuutoksen sijaan ohjaukseen syötetään yhden aikayksikön pituinen impulssi. Impulssivastekertoimet h_i saadaan ulostulon arvoista näytteenottohetkillä kuten kuvassa 4. [2]

Askelvaste- ja impulssivastekertoimien välillä vallitsee seuraavat riippuvuudet:

$$h_i = s_i - s_{i-1} \quad (19)$$

$$s_i = \sum_{j=1}^i h_j . \quad (20)$$

Yhtälöiden (19) ja (20) avulla voidaan ratkoa toisen mallin kertoimet toisen mallin kertoimien avulla.



Kuva 3. Impulssivastemalli [2]

5.4 Dynamic Matrix Control

Yksi yleisimmistä MPC-algoritmeista on DMC. DMC:n kehitti Shell Oil Company 1960- ja 70-luvulla. DMC:n käyttöönoton voi tiivistää kolmeen toimenpiteeseen [2]:

1. Luodaan askelvastemalli, jonka pituus on N ja näytteenottoaika on Δt .
2. Määritellään ennustehorisontti P ja säätöhorisontti M ($N \geq P \geq M$)
3. Määritellään ohjauksen painokerroin W .

Seuraavassa esitellään DMC-algoritmin laskentaa lähteiden [2] ja [3] pohjalta.

Koska DMC perustuu askelvastemalliin, se voidaan esittää muodossa

$$\hat{y}_k = s_1 \Delta u_{k-1} + s_2 \Delta u_{k-2} + \cdots + s_{N-1} \Delta u_{k-N+1} + s_N \Delta u_{k-N}, \quad (21)$$

jonka yleinen muoto on

$$\hat{y}_k = \sum_{i=1}^{N-1} s_i \Delta u_{k-i} + s_N u_{k-N}. \quad (22)$$

\hat{y}_k on mallin prediktio aika-askeleella k
 u_{k-N} on käytetyt ohjaukset N askelta menneisyydessä.

On huomioitava, että aika-askeleella k mallin ulostulo poikkeaa suurella todennäköisyydellä todellisesta prosessiulostulosta. Tätä mitatun ulostulon ja mallin prediktion välistä eroa kutsutaan additiiviseksi häiriöksi d_k :

$$d_k = y_k - \hat{y}_k. \quad (23)$$

Kun häiriön suuruus on määritetty, voidaan mallin prediktio korjata sen avulla:

$$\hat{y}_k = \hat{y}_k + d_k. \quad (24)$$

Vastaavasti voidaan laskea korjattu prediktio ensimmäiselle aika-askeleelle tulevaisuuteen (\hat{y}_{k+1}^c) kaavasta

$$\begin{aligned} \hat{y}_{k+1}^c &= \hat{y}_{k+1} + \hat{d}_{k+1} \\ &= \sum_{i=1}^{N-1} s_i \Delta u_{k-i+1} + s_N u_{k-N+1} + \hat{d}_{k+1} \\ &= s_1 \Delta u_k + \sum_{i=1}^{N-2} s_i \Delta u_{k-i+1} + s_N u_{k-N+1} + \hat{d}_{k+1}. \end{aligned} \quad (25)$$

Tämän perusteella aika-askeleelle j tulevaisuudessa saadaan

$$\begin{aligned} \hat{y}_{k+j}^c &= \hat{y}_{k+j} + \hat{d}_{k+j} \\ &= \underbrace{\sum_{i=1}^j s_i \Delta u_{k-i+j}}_{\text{tulevien ohjausten vaikutus}} + \underbrace{\sum_{i=j+1}^{N-1} s_i \Delta u_{k-i+j} + s_N u_{k-N+j}}_{\text{menneiden ohjausten vaikutus}} + \underbrace{\hat{d}_{k+j}}_{\text{korjaus-termi}}. \end{aligned} \quad (26)$$

Tästä yhtälöstä voidaan erottaa käytettyjen ja tulevien ohjausten vaikutukset.

Yleisin oletus, ns. ”vakio-additiivinen-häiriöoletus”, on, että korjaustermi on tulevaisuudessa vakio:

$$\hat{d}_{k+j} = \hat{d}_{k+j-1} = \dots = d_k = y_k - \hat{y}_k. \quad (27)$$

On myös huomioitava, että ohjausmuutoksia ei ole säätöhorisontin M jälkeen.

DMC-ohjausta luodessa täytyy valita ennustehorisontin pituus P ja säätöhorisontin pituus M . Näiden valinnan jälkeen saadaan algoritmille matriisiesitys

$$\begin{aligned} \underbrace{\begin{bmatrix} \hat{y}_{k+1}^c \\ \hat{y}_{k+2}^c \\ \vdots \\ \hat{y}_{k+j}^c \\ \vdots \\ \hat{y}_{k+P}^c \end{bmatrix}}_{\substack{Px1 \\ \text{korjatut} \\ \text{ennusteet, } \hat{y}^c}} &= \underbrace{\begin{bmatrix} s_1 & 0 & 0 & \dots & 0 & 0 \\ s_2 & s_1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ s_j & s_{j-1} & s_{j-2} & \dots & \dots & s_{j-M+1} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ s_P & s_{P-1} & s_{P-2} & \dots & \dots & s_{P-M+1} \end{bmatrix}}_{\substack{PxM \\ \text{dynaaminen matriisi, } S_f}} \underbrace{\begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+M-2} \\ \Delta u_{k+M-1} \end{bmatrix}}_{\substack{Mx1 \\ \text{nykyiset ja} \\ \text{tulevat ohjaukset, } \Delta u_f}} \\ &+ \underbrace{\begin{bmatrix} s_2 & s_3 & s_4 & \dots & s_2 & s_2 \\ s_3 & s_4 & s_5 & \dots & s_2 & 0 \\ \vdots & \vdots & \vdots & \dots & 0 & 0 \\ s_{j+1} & s_{j+2} & \dots & s_{N-1} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{P+1} & s_{P+2} & \dots & 0 & \dots & 0 \end{bmatrix}}_{\substack{Px(N-2) \\ \text{matriisi, } S_p}} \underbrace{\begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+M-2} \\ \Delta u_{k+M-1} \end{bmatrix}}_{\substack{(N-2) \times 1 \\ \text{menneet} \\ \text{ohjaukset, } \Delta u_{past}}} \\ &+ s_N \underbrace{\begin{bmatrix} \Delta u_{k-N+1} \\ \Delta u_{k-N+2} \\ \vdots \\ \Delta u_{k-N+P} \end{bmatrix}}_{\substack{Px1 \\ \text{menneet} \\ \text{sisääntulot, } \Delta u_p}} + \underbrace{\begin{bmatrix} d_{k+1} \\ d_{k+2} \\ \vdots \\ d_{k+P} \end{bmatrix}}_{\substack{Px1 \\ \text{ennustetut} \\ \text{häiriöt, } \hat{d}}}, \quad (28) \end{aligned}$$

joka voidaan esittää matriisivektorimuodossa

$$\underbrace{\hat{y}^c}_{\substack{\text{korjatut} \\ \text{ulostulo-} \\ \text{ennusteet}}} = \underbrace{S_f \Delta u_f}_{\substack{\text{nykyisten} \\ \text{ja tulevien} \\ \text{ohjausten} \\ \text{vaikutus}}} + \underbrace{S_{past} \Delta u_{past} + s_N u_P}_{\substack{\text{menneiden ohjausten} \\ \text{vaikutus}}} + \underbrace{\hat{d}}_{\substack{\text{ennustetut} \\ \text{häiriöt}}}. \quad (29)$$

Tässä korjatun prediktion yhtälössä (29) ovat mukana ns. ”pakotetut” ja ”vapaat” vasteet. Pakotetuilla vasteilla tarkoitetaan nykyisten ja tulevien ohjausmuutosten vaikutusta ja vapailla tarkoitetaan ennustettuja muutoksia, jos ohjausmuutoksia ei ole tulevaisuudelle. Asetusarvon r ja prediktion välinen ero on

$$\underbrace{r - \hat{Y}^c}_{\substack{\text{korjatut} \\ \text{virhe-} \\ \text{ennusteet, } E^c}} = \underbrace{r - [S_{past}\Delta u_P + S_N u_P]}_{\text{pakottamaton virhe, } E} - S_f \Delta u_f, \quad (30)$$

joka voidaan tiivistää muotoon

$$E^c = E - S_f \Delta u_f. \quad (31)$$

Tässä ennustetut virheet koostuvat vapaasta vasteesta E ja pakotetusta vasteesta $-S_f \Delta u_f$.

Aiemmin johdetun pienimmän neliösumman menetelmän kustannusfunktion sovitusta tähän on

$$\Phi = \sum_{i=1}^P (e_{k+i}^c)^2 + w \sum_{i=0}^{M-1} (\Delta u_{k+i})^2. \quad (32)$$

Kustannusfunktion kaksi summalauseketta voidaan esittää matriisivektori muodossa

$$\begin{aligned} \sum_{i=1}^P (e_{k+i}^c)^2 &= [e_{k+1}^c \quad e_{k+2}^c \quad \dots \quad e_{k+P}^c] \begin{bmatrix} e_{k+1}^c \\ e_{k+2}^c \\ \vdots \\ e_{k+P}^c \end{bmatrix} \\ &= (E^c)^T E^c \end{aligned} \quad (33)$$

ja

$$\begin{aligned} w \sum_{i=0}^{M-1} (\Delta u_{k+i})^2 &= w \cdot [\Delta u_k \quad \Delta u_{k+1} \quad \dots \quad \Delta u_{k+M-1}] \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+M-1} \end{bmatrix} \\ &= \Delta u_f^T W \Delta u_f \end{aligned} \quad (34)$$

Tämän perusteella kustannusfunktio voidaan kirjoittaa muotoon

$$\Phi = (E^c)^T E^c + \Delta u_f^T W \Delta u_f. \quad (35)$$

Sijoittamalla tähän korjatun virheen E^c lauseke saadaan

$$\Phi = (E - S_f \Delta u_f)^T (E - S_f \Delta u_f + (\Delta u_f)^T W \Delta u_f). \quad (36)$$

Ratkaisemalla yhtälö (36) saadaan ohjausmuutokseksi

$$\Delta u_f = \underbrace{(S_f^T S_f + W)^T S_f^T}_K \cdot E. \quad (37)$$

Tässä K eli ohjauksen vahvistusmatriisi on vakio, joten ohjausmuutos Δu_f on suoraan suhteessa virhevektoriin E .

Koska vain ensimmäinen ohjausmuutos käytetään, saadaan ohjausmuutoksen yhtälö muotoon

$$\Delta u_k = K_1 E. \quad (38)$$

K_1 on vahvistusmatriisi K :n ensimmäinen rivi.

5.4.1 Viritysparametrien vaikutus

Mallin pituus N ja näyteaika t riippuvat prosessista. N :ksi valitaan suunnilleen prosessin asettumisaika askelmuutoksen jälkeen, joka on useimmilla prosesseilla noin 50 askelvastekertoimen mittainen. Näyteaika t valitaan usein prosessissa käytetyn perusaikayksikön kymmenesosaksi. [2; 3]

Ennuste- ja säätöhorisonttien pituus poikkeaa toisistaan. Ennustehorisontti valitaan usein paljon pidemmäksi kuin säätöhorisontti. Tämä pätee varsinkin, jos ohjauksen painokerroin w on nolla. Useimmiten pidempi ennustehorisontti P vähentää ohjausvirhettä mallin heikkouden takia. Esimerkiksi usein P on 20 ja M on 1-3. [2; 3]

Ohjauksen painokerroin valitaan usein nolllaksi, jos ennustehorisontti on huomattavasti pidempi kuin säätöhorisontti. Säättöhorisontin kasvaessa ohjauksella on tapana muuttua aggressiivisemmaksi, jolloin painokerrointa tarvitaan hillitsemään ohjausmuutoksia. [2; 3]

6 Mallin vaikutuksen testaaminen

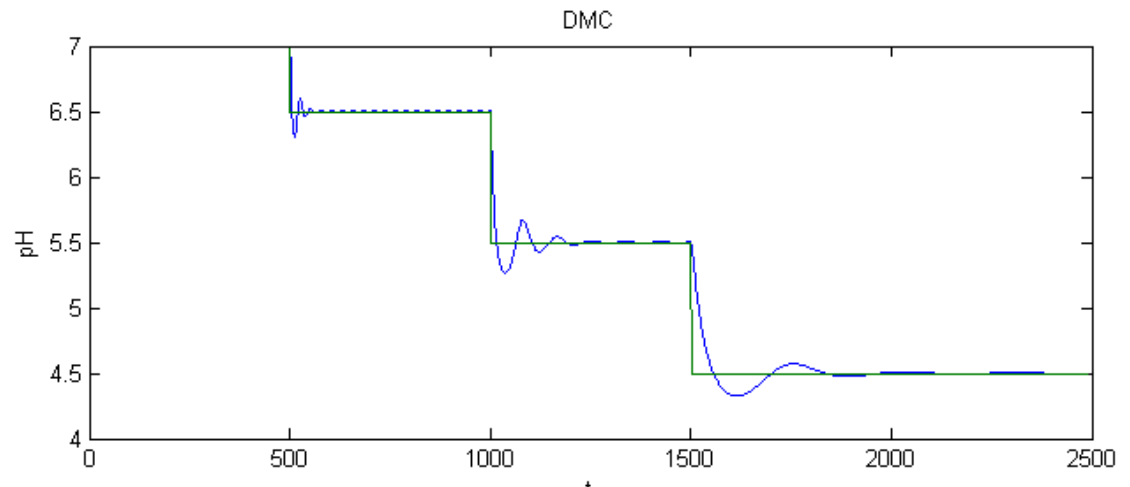
Lineaarista MPC-säätömenetelmää käytettäessä logaritmisen (eli epälineaarisen) pH:n säätämiseen on hyvin tärkeää valita oikein, mille pH-alueelle malli luodaan. Yhdessä toimintapisteessä tehty malli ei välttämättä toimi toisessa, ja väärän kokoisella askeleella luotu askelvastemalli voi hidastaa tai jopa estää prosessin toiminnan halutulla alueella. Tässä kappaleessa esitetään havainnollistavien kuvien avulla, millaisia vaikutuksia mallin laadulla on ohjauksen toimivuuteen.

6.1 Toimintapiste

Mallia luotaessa on pH:n tapauksessa tärkeää valita toimintapiste tarkasti ohjauksen optimaalisen toiminnan takaamiseksi.

Jos malli luodaan pH-alueen 6-8 ulkopuolella, on hyvin todennäköistä, että säätö ei tule toimimaan pH-asteikon herkimmällä alueella. Tämä johtuu siitä, että muutosherkkyys pH-alueen 6-8 ulkopuolella on huomattavasti pienempi, jolloin tarvitaan suurempaa vahvistusta ohjaukselle. Suurempi vahvistus puolestaan tarkoittaa, että herkällä alueella oltaessa ohjaus on liian suuri eikä prosessi pääse tasapainotilaan.

Lähellä arvoa $\text{pH} = 7$ luodut mallit puolestaan toimivat hyvin koko herkällä alueella 6-8, mutta herkkyyden vuoksi ohjauksen vahvistus on hyvin pieni. Tästä seuraa, että prosessin toiminta hidastuu herkän alueen ulkopuolelle mentäessä (kuva 5).



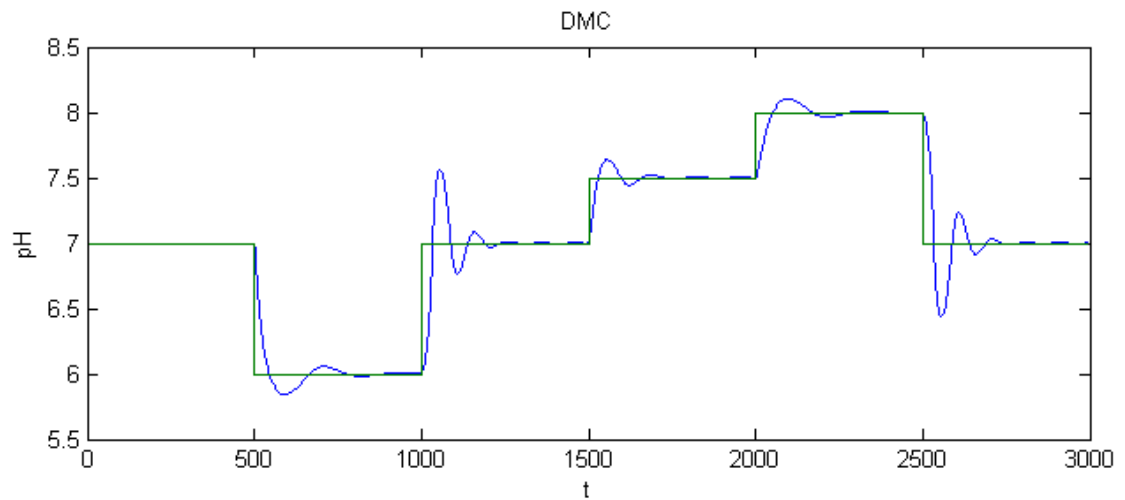
Kuva 4. Ohjauksen hidastuminen.

6.2 Askelvastemallin askel

Askelvastemallia luotaessa on hyvin tärkeää valita askeleen suuruus huolella. Askeleen koko vaikuttaa suoraan mallin toimivuuteen ja ohjauksen nopeuteen. Seuraavassa esitetään askelkoon vaikutuksia luotaessa malli toimintapisteessä pH 7.

6.2.1 Liian pieni askel

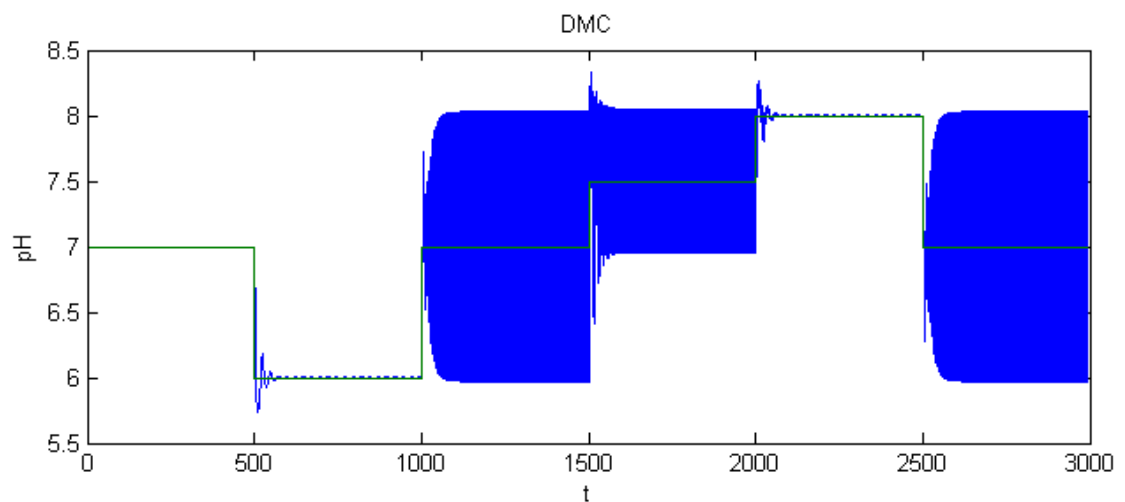
Liian pienen askeleen käyttö mallia luotaessa voi johtaa prosessin hidastumiseen (kuva 6). Toimintapisteen pH 7 tapauksessa liian pieni askel (pH-muutos alle 0,3) tekee ohjauksen vahvistuskertoimesta äärimmäisen pienen neutraalitalan herkkyyden takia. Pienen vahvistuksen vuoksi ohjaus hidastuu jo alueen $\text{pH} = 6,5 \dots 7,5$ ulkopuolella, jolloin optimaalisesti toimiva alue jää hyvin pieneksi ja epäkäytännölliseksi.



Kuva 5. Liian pienen askeleen vaikutus.

6.2.2 Liian suuri askel

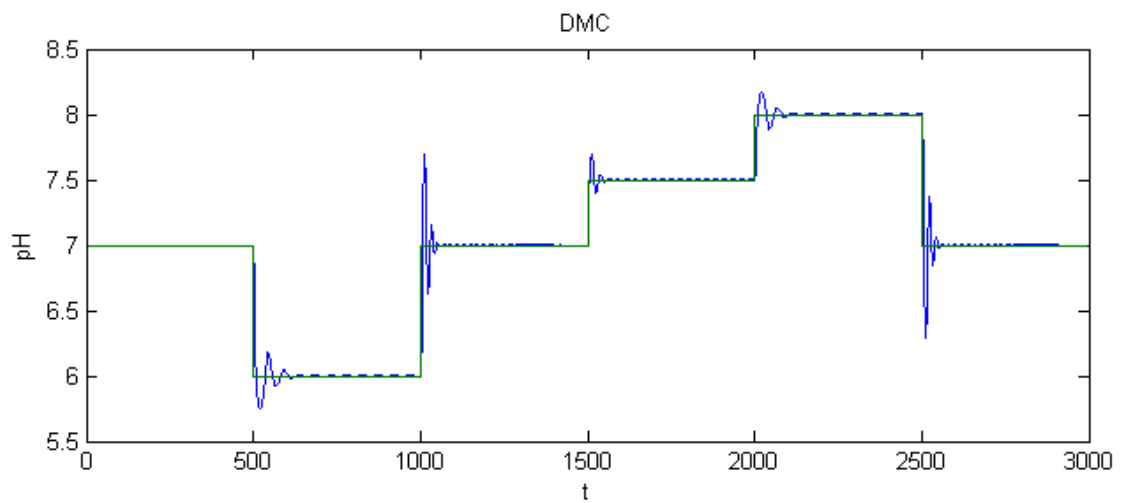
Askeleen suuruutta valittaessa on erityisen tärkeää olla valitsematta liian suurta askelta. Liian suurella askeleella on suurempi haittavaikutus kuin liian pienellä askeleella. Liian suuri askel (pH-muutos yli 2) tuo ohjaukseen suuren vahvistuksen, mikä aiheuttaa herkästi prosessin toimimattomuuden herkillä muutosalueella arvon $\text{pH} = 7$ ympärillä (kuva 7).



Kuva 6. Liian suuren askeleen vaikutus.

6.2.3 Optimaalinen askel

Sopivankokoinen askel tuo suunnilleen yhden pH-yksikön suuruisen muutoksen prosessiin. Tällöin ohjauksessa käytetty vahvistus on skaalattu sopivankokoiseksi herkällä pH-alueella 6-8. Hyvin valitun askeleen tuottamalla mallilla prosessi toimii optimaalisesti tavoitealueella, mutta toimii myös alueen ulkopuolella, vaikkakin sitä hitaammin, mitä kauempana optimaalisesta alueesta ollaan. Kuvassa 8 on esitetty ohjauksen toiminta hyvin tehdyllä mallilla.



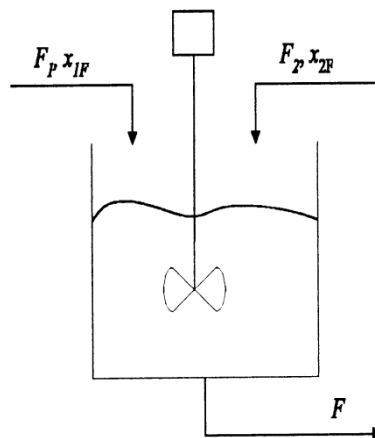
Kuva 7. Optimaalisella mallilla toimiva ohjaus.

7 pH-säätöpiirin toteutus

Tässä työssä päädyttiin tutkimaan lineaarisen MPC:n toimivuutta yksinkertaisen pH-prosessimallin ohjauksessa. MPC-algoritmina käytettiin DMC:tä ja mallina yksinkertaista askelvastemallia toimintapisteen pH 7 ympäristössä. Verrattavaksi säätömenetelmäksi rakennettiin PI-säätöpiiri ja viritettiin se Cohen-Coon-menetelmällä. Kaikki laskenta ja simulointi toteutettiin Matlab-ohjelmiston funktioilla ja Simulinkillä. Seuraavaksi esitetään lyhyesti simulaatioiden osat, joiden tarkemmat koodirakenteet ja kaaviot on esitetty liitteissä 1-8.

7.1 pH-prosessi

pH-prosessia kuvataan yksinkertaisella säiliöllä (kuva 9), johon happoliuosta syötetään vakiovirtana ja emäsliuoksen virtausta ohjataan säätimellä. Säiliössä olevan nesteen määrä pysyy vakiona, kun ulosvirtaavan nesteen määrä on sama kuin sisään tulevien summa. Prosessissa on oletuksena, että kaikki muutokset tapahtuvat ilman viiveitä.



Kuva 8. pH-prosessi [1].

Prosessin simulointia varten Simulinkissä luotiin oma S-funktio, jonka sisääntulona on säätimeltä tuleva ohjaus. Prosessifunktiolle asetettiin erillisessä parametritiedostossa

happo- ja emäsluosten pH-arvot sekä happoliuoksen sisääntulovirtauksen määrä. pH-prosessifunktion Matlab-koodi on esitetty liitteessä 1.

7.2 DMC-säädin

DMC-säätimen luomiseen käytettiin muokattua versiota Yi Caon [15] kirjoittamasta Simulink DMC -säätimestä. Tämä säädinfunktio laskee ohjausmuutoksen prosessille aiemmin luvussa 5.4 esitetyn DMC-algoritmin mukaisesti. Sisääntuloina säätimelle tuodaan nykyhetken mittaustulos prosessista ja nykyinen asetusarvo, ulostulona säädin antaa ohjausarvon prosessille. DMC-funktion koodi löytyy liitteestä 2.

DMC-laskennassa tarvittava malli luodaan erillisellä askelvastesimulaattorilla, jonka Simulink-kaavio on esitetty liitteessä 3. Tämä piiri simuloi viisi askelvastekoetta pH-prosessiin ja tallentaa ulostulot muuttuinaan. Ennen varsinaista säätösimulointia askelvastekokeen tulokset täytyy jakaa käytettyjen askelten suuruudella ja laskea saatujen askelvasteiden keskiarvo. Tässä työssä mallin viimeistelyyn tehtiin oma funktio, joka suorittaa edellä mainitun toimenpiteen, minkä jälkeen malli on valmis prosessisimulointia varten.

7.3 PI-säädin

Tässä työssä käytetty PI-säädin perustuu aiemmin esiteltyyn PID-algoritmiin ilman D-termiä. Säätimen parametrit laskettiin Cohen-Coon-menetelmällä samasta askelvastemallista, jota DMC-säädin käyttää. PI-säätimen vahvistusta piti kuitenkin laskea huomattavasti, koska prosessi värähteli liikaa toimintapisteessä perusvirityksellä. PI-säätimen koodi seuraa liitteenä 4.

7.4 DMC- ja PI-säätimien vertailu

Säätimien välinen vertailutesti suoritettiin hyvän mallin mukaan viritetyillä parametreilla, jotka on esitetty taulukossa 3.

Taulukko 3. Säätimien parametrit

PI-säädin		DMC-säädin	
K_C	0,003	P	5
K_I	2,703	M	1
		W	1

Säätimien vertailuun tehtiin asetusarvomuutos- ja häiriötilannetestejä.

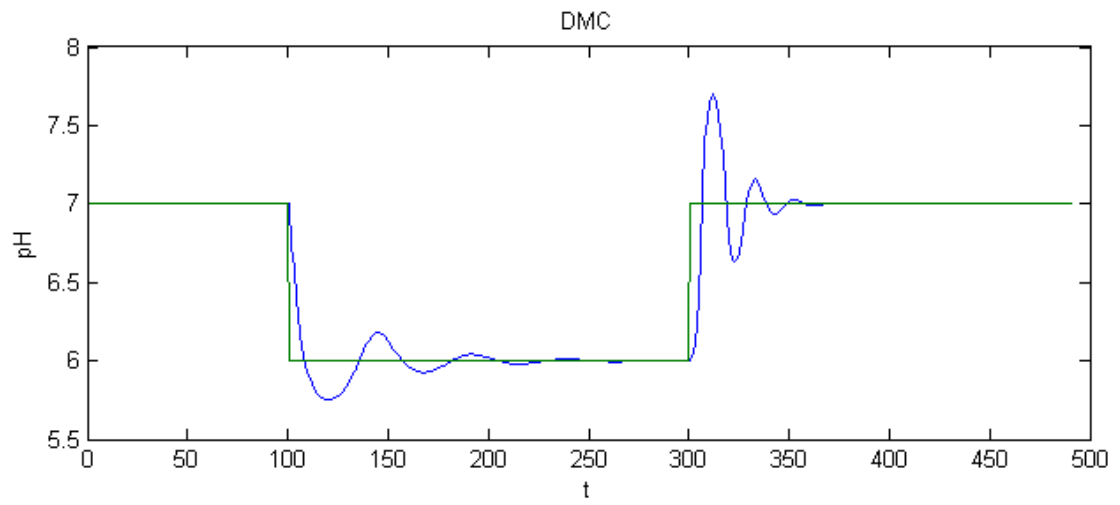
7.4.1 Testi 1: Asetusarvomuutokset

Ensimmäisessä testissä verrataan säädinten reagointia kahteen erisuuruiseen asetusarvomuutokseen (taulukot 4 ja 5) tutkien prosessien muutosta ja palautumista takaisin alkuperäiseen asetusarvoon (kuvat 10 ja 11). Jokaisessa testiosuudessa prosessi alkaa tasapainotilassa pH 7.

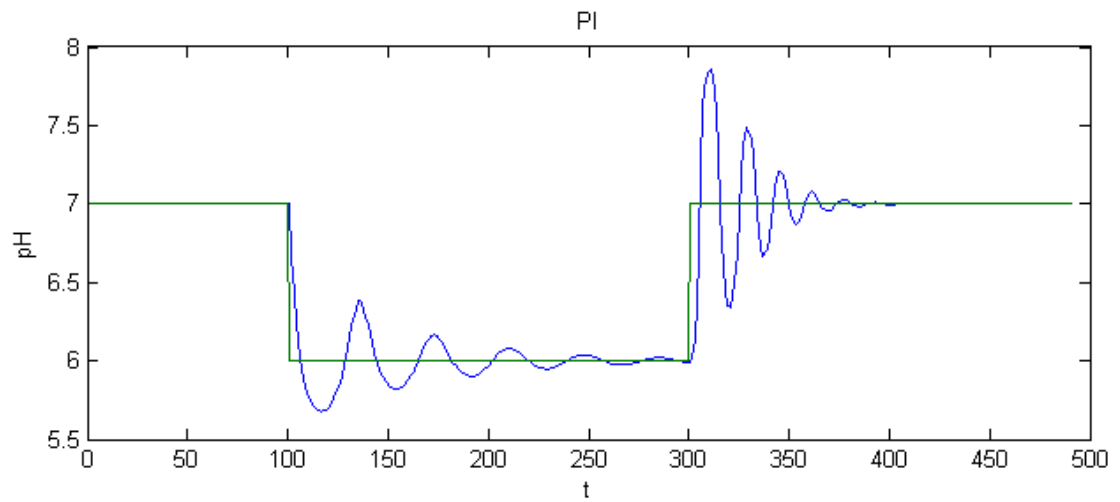
Muutos 1

Taulukko 4. Ensimmäisen muutoksen asetusarvot.

pH:n asetusarvo	Muutoshetki
6	100
7	300



(a)



(b)

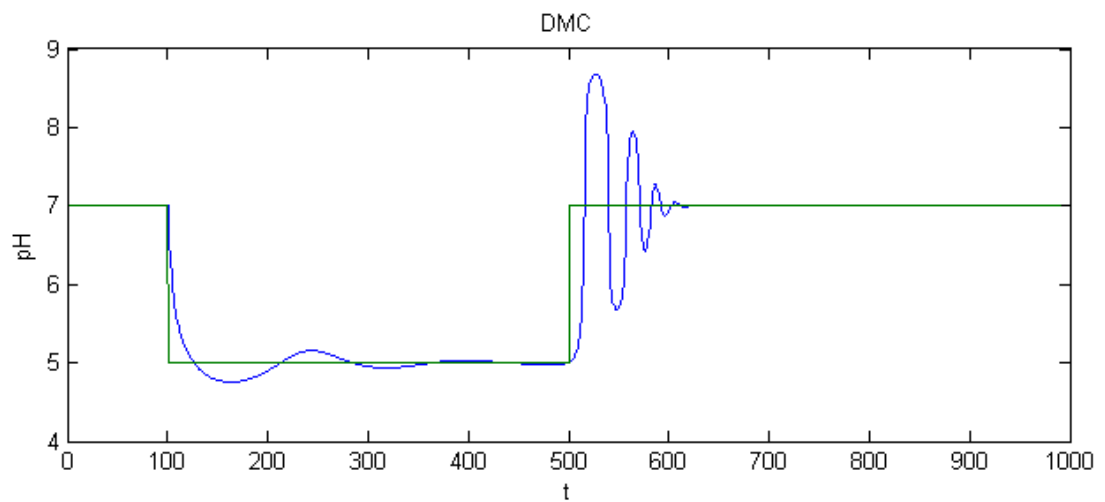
Kuva 10. Ensimmäisen muutostestin toiminta (a) DMC-säätimellä ja (b) PI-säätimellä.

Kuvasta 10 huomataan, että PI-säätimellä menee huomattavasti kauemmin saavuttaa asetusrvon määräämä tila verrattuna DMC-säätimeen. On myös huomattavan suuri ero säädinten oskilloinnissa asetusrvon ympärillä.

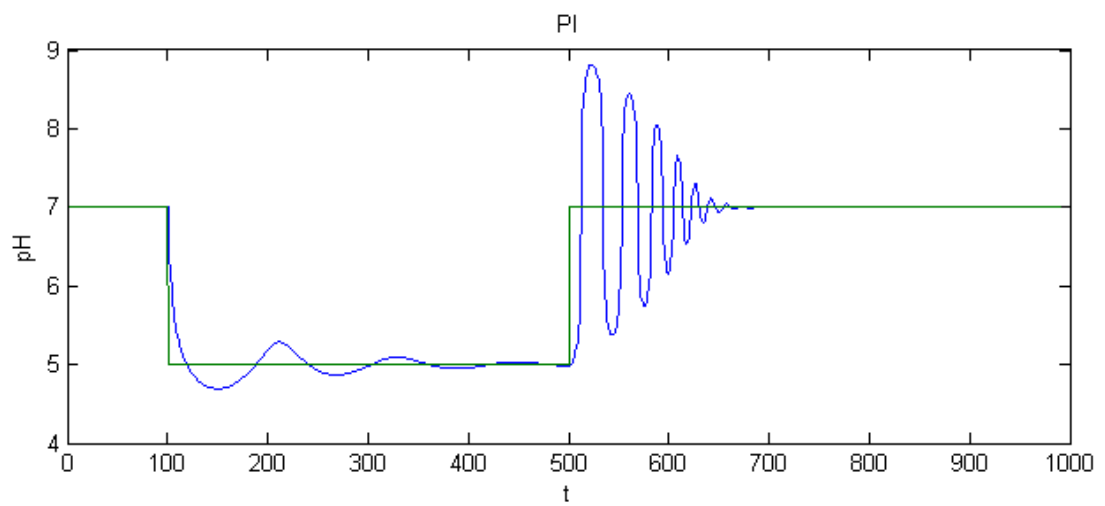
Muutos 2

Taulukko 5. Toisen muutoksen asetusrvot.

pH:n asetusarvo	Muutoshetki
5	100
7	500



(a)



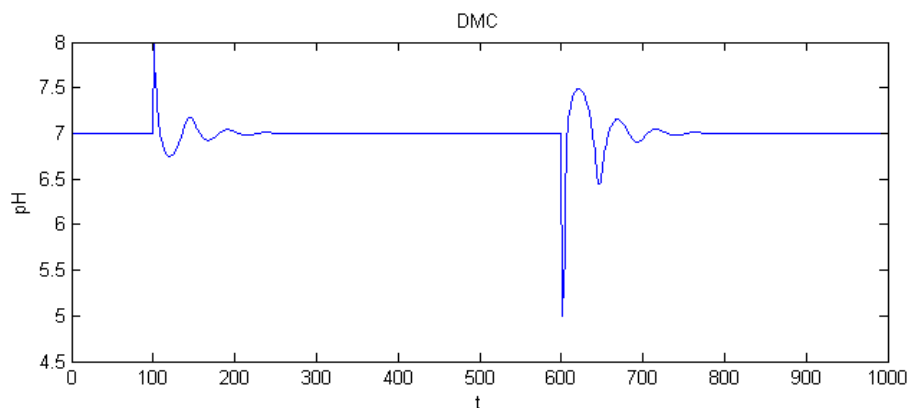
(b)

Kuva 11. Toisen muutostestin toiminta (a) DMC- ja (b) PI-säätimellä.

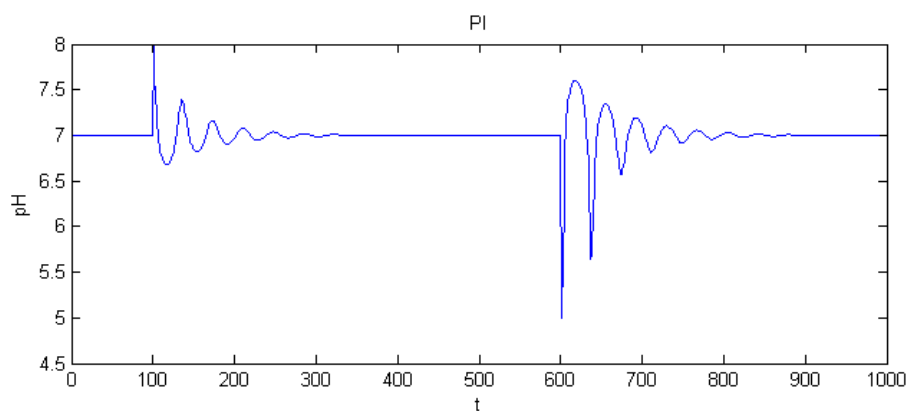
Kuvasta 11 huomataan, että suuremmilla asetusarvomuutoksilla säätimien tehokkuus laskee huomattavasti. Tämä johtuu siitä, että pH on hyvin epälineaarinen suure näiden kahden asetusarvon välillä. Molemmat säätimet on viritetty toimimaan hyvin alueella $\text{pH} = 6 \dots 8$, joten kyseisen alueen ulkopuolella ohjaus heikkenee. Yleinen ero PI- ja DMC-säätimien välillä on selvä tässäkin testissä: PI-säätimellä kestää kauemmin saavuttaa tavoitearvo ja sillä on taipumus oskilloida pidempään.

7.4.2 Testi 2: Häiriökorjaus

Vertailun toisessa testissä pH-prosessiin syötetään kaksi erisuuruista häiriösignaalia ja tarkastellaan säätimien kykyä palauttaa prosessi tavoitetilaa $\text{pH} 7$. Ensimmäinen häiriö nostaa pH-arvoa yhdellä ja toinen laskee kahdella yksiköllä.



(a)



(b)

Kuva 12. Häiriötesti (a) DMC- ja (b) PI-säätimellä.

Kuvissa 12 (a) ja (b) huomataan samoja seikkoja, kuin ensimmäisessä testissä. PI-säätimellä menee yleisesti kauemmin palautua tavoitearvoon ja oskillointia tapahtuu huomattavasti enemmän kuin DMC-säätimellä.

7.5 Tulokset

Tehtyjen testien tuloksista käy selväksi, että DMC-säädin suoriutuu pH-säädöstä paremmin kuin PI-säädin. Tämä seikka todetaan monissa aiheeseen liittyvissä tutkimuksissakin [6; 9; 10]. Näissä tutkimuksissa erot käyvät simulointeja paremmin selviksi, koska kokeet on suoritettu oikeissa testiympäristöissä.

Tässä työssä tehtyjen testien perusteella huomataan, kuinka tärkeää epälineaarista prosessia säädettäessä on valita toiminta-alue tarkasti lineaarisilla menetelmillä. Ohjauksesta saadaan suhteellisen hyvä, jos rajataan haluttu toiminta-alue tarkasti ja säätimet viritetään sen mukaisesti. Huonosti valittu toiminta-alue tuo prosessiohjaukseen hitautta ja turhaa oskillointia.

Oikeaa pH-prosessia säädettäessä suoriutumiseroa kasvattaa prosessin kuollut aika, jonka käsittelyyn DMC-säädöllä on huomattavasti paremmat edellytykset kuin PI-säätimellä. PI-säädin voidaan viritellä ottamaan huomioon kuollut aika (Cohen-Coon-menetelmä), mutta DMC-säädössä kuollut aika tulee mukaan laskennassa käytetyssä mallissa.

8 Yhteenveto

Tässä työssä on esitelty teoriaa hapoista, emäksistä, pH:sta, pH-mittauksesta ja sen säädöstä. Esitetyn teorian perusteella on luotu säätöpiirit yksinkertaiselle pH-prosessille ja analysoitu niiden tuottamia tuloksia.

Säätöpiireille tehtyjen askelmuutos- ja häiriötestien perusteella voitiin todeta, että DMC-säädin soveltuu paremmin pH-säätöön kuin PI-säädin. Molemmat säätimet tosin tuottivat kohtuulliset tulokset hyvin viritettyinä, mutta sovellettaessa oikeaan pH-prosessiin PI-säätimen toimivuus heikkenee enemmän kuin DMC-säätimen.

Tätä työtä olisi voitu laajentaa kattamaan myös epälineaariseen säätöön kehitettyjen NMPC (Non-linear MPC)- ja WMPC (Wiener MPC) -säädinten toiminta. Näistä menetelmistä jälkimmäinen on monien tutkijoiden [6; 9; 10] mielestä lineaarista MPC:tä parempi pH:n säätömenetelmä, mutta sen käyttöönotto vaatii enemmän tietoa prosessitekniikasta ja siihen liittyvästä matematiikasta kuin tässä työssä käytetyt säätömenetelmät.

Tässä työssä luotuja säätimiä voidaan suhteellisen helposti testata tulevassa oppimisympäristössä. Säätöpiireissä käytettyjen pH-prosessiblokkien tilalle tulee vain kytkeä itse prosessilaitte OPC (OLE for process control) -yhteyden avulla.

Lähteet

- 1 Ali, E. pH control using PI control algorithms with automatic tuning method. Trans IChemE, Vol 79, Part A, pp. 611-620, July 2001
- 2 Bequette, B. Wayne. Process Control. Modeling, Design, and Simulation. Upper Saddle River, New Jersey: Pearson Education Inc. 2003.
- 3 Brosilow, Coleman & Joseph, Babu. Techniques of Model-based Control. Upper Saddle River, New Jersey: Prentice Hall PTR, A Division of Pearson Education Inc. 2002.
- 4 Falck, Per. Lukioasteen kemia 1. Helsinki: Kirjayhtymä. 1972.
- 5 Falck, Per. Lukioasteen kemia 2. Helsinki: Kirjayhtymä. 1973.
- 6 Gómez, J.C., Jutan A., and Baeyens, E. Wiener model identification and predictive control of a pH neutralisation process. IEE Proc.-Control Theory Appl., Vol. 151, No.3, pp. 329-338, May 2004.
- 7 Katebi, M.R. & Moradi, M.H. Predictive PID controllers. IEE Proc.-Control Theory Appl., Vol. 148, No. 6, pp. 478-487, November 2001.
- 8 Lehtonen Pekka O. Potentiometrinen analyysi. *pH- ja ISE-mittaukset*. AEL-ammattitieto. Helsinki: Oy Edita Ab. 1998.
- 9 Norquay, Sandra J., Palazoglu, Ahmet, and Romagnoli José Alberto. Model predictive control based on Wiener models. Chemical Engineering Science, Vol. 53, No. 1, pp. 75-84, 1998.
- 10 Norquay, Sandra J., Palazoglu, Ahmet, and Romagnoli José Alberto. Application of Wiener Model Predictive Control (WMPC) to a pH neutralization experiment. IEEE transactions on control systems technology, Vol. 7, No. 4, pp. 437-445, July 1999.
- 11 Pihkala, Juhani. Prosessisuureiden mittaustekniikka. Opetushallitus. Helsinki: Oy Edita Ab. 1997.
- 12 Ylén, Jean-Peter. Measuring, modelling and controlling the pH value and the dynamic chemical state. Helsinki University of Technology, Control Engineering Laboratory, Report 127. Espoo: 2001.
- 13 PID controller, 2010. (WWW-dokumentti.) Wikipedia
< http://en.wikipedia.org/wiki/PID_controller >. Päivitetty 13.3.2010. Luettu 15.5.2010.
- 14 PID Tuning Via Classical Methods, 2006. (WWW-dokumentti.)
The Michigan Chemical Process Dynamics and Controls Open Text Book
< <http://controls.engin.umich.edu/wiki/index.php/PIDTuningClassical> >
Päivitetty 16.10.2007. Luettu 15.5.2010.
- 15 Cao, Yi. MPC Tutorial IV: DMC Simulink Block and Example, 2009. (WWW-dokumentti.) Matlab Central.
<<http://www.mathworks.com/Matlabcentral/fileexchange/25412-mpc-tutorial-iv-dmc-Simulink-block-and-example>>. Päivitetty 18.3.2010. Luettu 4.5.2010.

```

function [sys,x0] = pHprocess(t,x,u,flag)

% pH-prosessin simulointi

global sume

if abs(flag) == 1                % Derivate
    sys = dbeta(x,u);
elseif abs(flag) == 3            % Output
    sys = [x(1)];
elseif abs(flag) == 0            % Initialize
    sys = [1, 0, 1, 1, 0, 0];
    x0=0;
    sume=0;
else
    sys = [];
end

function deltab = dbeta(X,U)
%Hapon ja emäksen pitoisuudet ja virtaukset
X1F=10^-value(2);
X2F=10^(value(3)-14);
F1=value(4);
F2=U(1)+(0.1*10^-value(2))/10^(value(3)-14);
F=F1+F2;
%Säiliön pitoisuuden muutoksen laskeminen
deltab(1) = (F1*X1F - F2*X2F - F*X(1))/value(5);

function StopH = StopH(dbeta)
%Uuden pH arvon laskeminen pitoisuudesta
%Ch-beta*Ch-kw=0 => Ch = ?
for i=1:length(dbeta)
    kw = 1e-14;
    d = sqrt(dbeta(i)^2+4*kw);
    Ch = (dbeta(i)+d)/2;
    pH(i) = -log10(Ch);
end
StopH = pH;
    
```

```

function [sys,x0,str,ts] = dmcsfun(t,x,u,flag)
%DMCSFUN S-Function for Dynamic Matrix Control.
%By Yi Cao, Cranfield University, UK, (c) 2009
% Modified by Ismo Leszczynski

global SrModel
persistent K

switch flag,

    % Initialization %
    case 0,

        %Parameters
        Ts=0.1;                % Sampling time
        K.sr=SrModel;          % Step Response Model
        K.p=5;                 % P, prediction horizon
        K.m=1;                 % M, moving horizon
        K.v=[];                % History of control
        K.la=1;                % input weight, aka W
        K.a=0;                 % Reference smooth factor
        K.y=[];                % Initialization DMC
        K=dmc(K);

        sizes = simsizes;
        sizes.NumContStates = 0;
        sizes.NumDiscStates = 0;
        sizes.NumOutputs = 1;
        sizes.NumInputs = 2; %measured output and setpoint
        sizes.DirFeedthrough = 1;
        sizes.NumSampleTimes = 1;
        sys = simsizes(sizes);
        str = [];
        x0 = zeros(0,1);
        ts = [Ts 0];

    % Update %
    case 2,
        sys = x;

    % Output %
    case 3,
        K.y = u(1);            % measurement
        K.r = u(2);            % setpoint
        K = dmc(K);
        sys = K.u;

    case 9,
        sys = [];

```

```

    % Unexpected flags %
    otherwise
        error(['unhandled flag = ', num2str(flag)]);
end

function p=dmc(p)

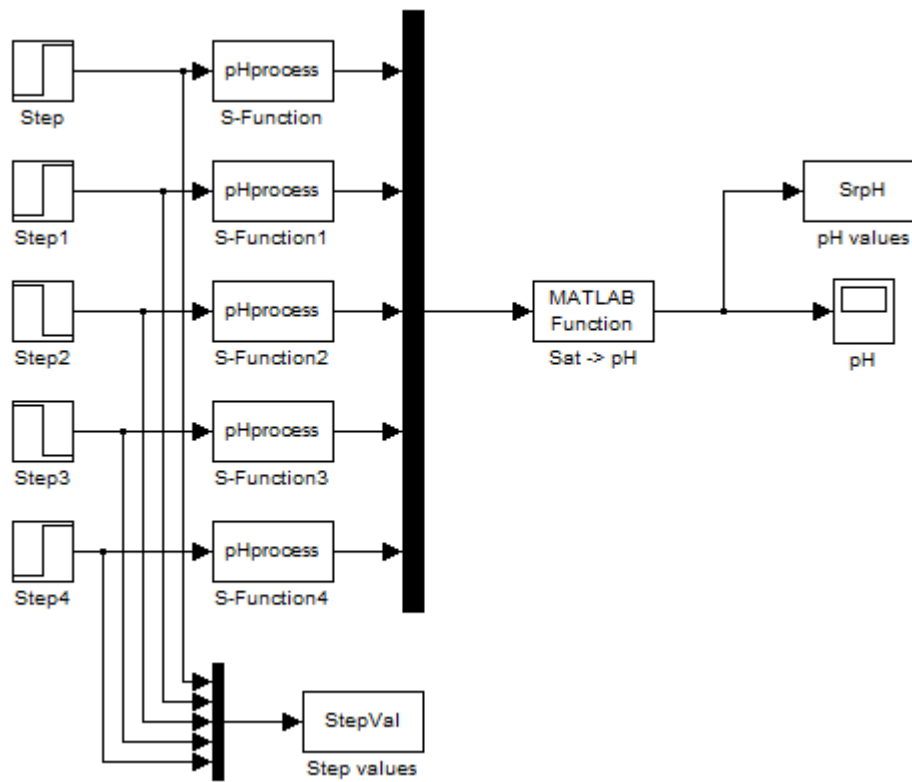
% Input and output check
error(nargchk(1,1,nargin));
error(nargoutchk(0,1,nargout));

% length of step response
N=numel(p.sr);
P=p.p;

% initial setup
if isempty(p.v)
    % number of past inputs to keep
    n=N-P;
    % storage for past input
    p.v=zeros(n,1);
    % matrix to calculate free response from past input
    x=p.sr(1:n);
    p.F=hankel(p.sr(2:P+1),p.sr(P+1:N))- repmat(x(:)',P,1);
    % dynamic matrix
    p.G=toeplitz(p.sr(1:P),p.sr(1)*eye(1,p.m));
    % calculate DMC gain
    R=chol(p.G'*p.G+p.la*eye(p.m));
    K=R\(R'\p.G');
    % only the first input will be used
    p.k=K(1,:);
    p.u=0;
end

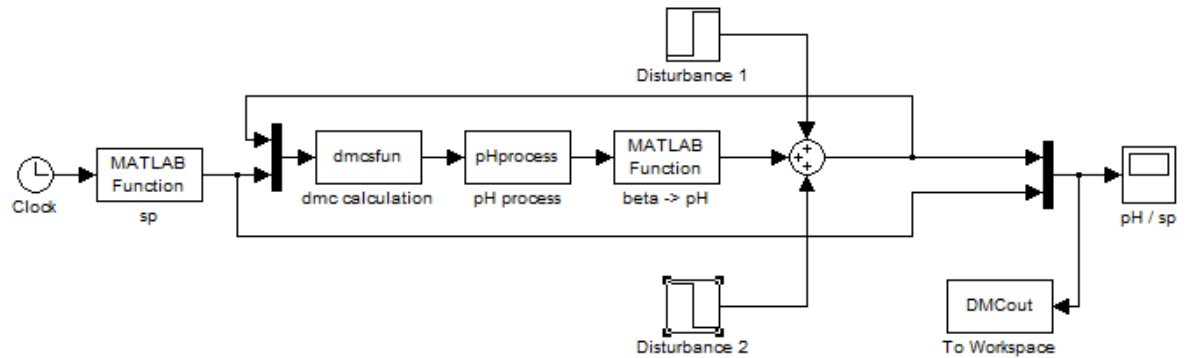
if isempty(p.y)
    return
end
% free response
f=p.F*p.v+p.y;
% smooth reference
nr=numel(p.r);
if nr>=P
    ref=p.r(1:P);
else
    ref=[p.r(:);p.r(end)+zeros(P-nr,1)];
end
w=filter([0 (1-p.a)], [1 -p.a], ref,p.y);
% DMC input change
u=p.k*(w-f);
% past input change for next step
p.v=[u;p.v(1:end-1)];
% next input
p.u=p.u+u(1);

```

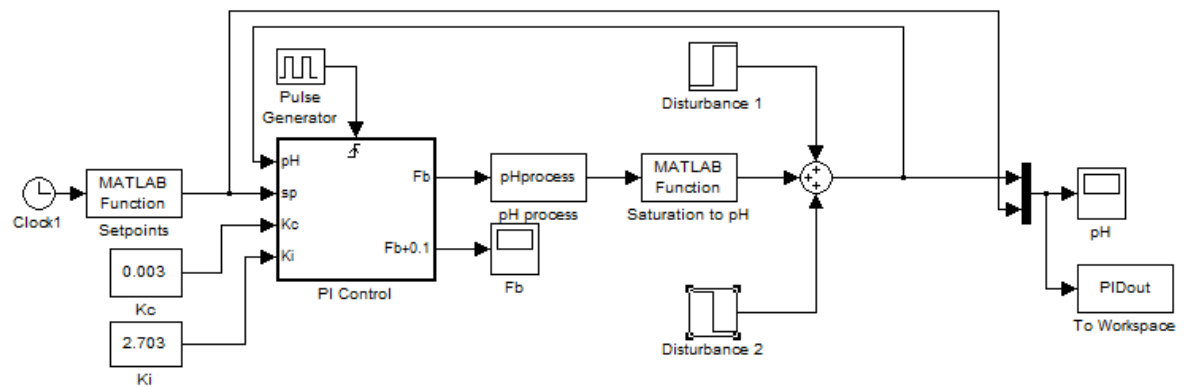



```
function [mv] = Ipid(p)
% PI-säädin
global sume
% Erosuureen laskenta
ero=p(2)-p(1);
% Ki- ja Kc-arvojen erotus sisääntulosta
Ki=p(4);
Kc=p(3);
sume = sume + ero;
% PI-säätimen ulostulolaskenta
mv = Kc*ero+Kc*Ki*sume;
```

DMC:



PI:



```
function sp = setp(t)
% Asetusarvojen määrittely, sisääntulo t on
% aika ja palautuksena pitää tulla aikaa vastaava
% asetusrvo.

s = 7;
if (t < 10)
    s = 7;
end

if (t >= 10 && t < 20)
    s = 7;
end

if (t >= 20 && t < 40)
    s = 7;
end

if (t >= 40 && t < 50)
    s = 7;
end

if (t >= 50)
    s = 7;
end
sp = s;
```

```
function SplineAndAvg = SAA(pH, StepVal)
%Sovitetaan käyrät Step-responsesta
%saaduille arvoille

%pH mallin pituus
x=1:size(pH,1);
%Käyrän sovituksen askelmäärä
xx=1:size(pH,1);

%Käyrien sovitus pisteille
for i=1:size(pH,2)
    Spl(:,i)=spline(x,pH(:,i),xx);
end

%Poikkeaman laskeminen aloitusarvosta,
%esim jos alkaa 7:sta ja arvo on 7,5 niin
%poikkeamaksi tulee 0,5.
for i=1:50
    Spl(i,1)=Spl(i,1)-pH(1,1);
    Spl(i,2)=Spl(i,2)-pH(1,2);
    Spl(i,3)=Spl(i,3)-pH(1,3);
    Spl(i,4)=Spl(i,4)-pH(1,4);
    Spl(i,5)=Spl(i,5)-pH(1,5);
end

%Summamatriisin alustus
SplSum=zeros(size(Spl,1));

%Käyrien arvojen jako Step arvolla ja summaaminen
for i=1:size(Spl,1)
    for j=1:size(Spl,2)
        SplSum(i)=SplSum(i)+Spl(i,j)/StepVal(j);
    end
end

%Summien keskiarvot
for i=1:length(SplSum)
    SplOut(i)=SplSum(i)/size(Spl,2);
end

SplineAndAvg = SplOut;
```

```
function valueReturn = value(x)
switch x
case 1
    %Säiliön alkutilan pH
    value = 7;
case 2
    %Happoliuoksen pH
    value = 3.2;
case 3
    %Emäsluoksen pH
    value = 10.5;
case 4
    %Happoliuoksen virtaus, Fa
    value = 0.1;
case 5
    %Säiliön tilavuus, V
    value = 2.0;
otherwise
    value = 0;
end

valueReturn = value;
```